2

AN ANALYSIS OF AIR FORCE FACILITY PROGRAMMING AND ITS EFFECT ON DESIGN AND CONSTRUCTION

THESIS

Michael A. Ross, Captain, USAF

AFIT/GEM/DEE/90S-14

01 1 25 034

AFIT/GEM/DEE/90S-14

AN ANALYSIS OF AIR FORCE FACILITY PROGRAMMING AND ITS EFFECT ON DESIGN AND CONSTRUCTION

THESIS

Michael A. Ross, Captain, USAF

AFIT/GEM/DEE/90S-14

The opinions and conclusions in this paper are those of the author and are not intended to represent the official position of the DOD, USAF, or any other government agency.

AFIT/GEM/DEE/90S-14

AN ANALYSIS OF AIR FORCE FACILITY PROGRAMMING AND ITS EFFECT ON DESIGN AND CONSTRUCTION

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Technical Management

Michael A. Ross, B.S.

Captain, USAF

September 1990

Preface

The purpose of this study was to find a way to improve how the Air Force designs and constructs facility projects. Past research indicated that one problem effecting quality facilities was programming. Also, programming is the beginning of the design and construction process, a logical place to start the research.

The research was conducted using the Delphi Method. The Delphi technique is a research method that relies on the judgment of "experts." My research involved two panels of "experts": (1) professional programmers outside the Air Force, and (2) Air Force Chief Engineers in Civil Engineering organizations. The two groups allowed me to compare programming practices and attitudes. From my conclusions, I proposed a new programming model designed to solve current problems, and take advantage of "good" programming practices.

I am deeply indebted to two people. First, I thank my thesis advisor, Captain Don Colman, for his patience and encouragement. Second, I acknowledge Carol Ross Barney. As my sister and professional architect, she has been a great source of support and information. Finally, a word of thanks to my "experts." Without their time and experience, the research was not possible.

Michael A. Ross

Table of Contents

List of Figures

List of Tables

Table	Page

Table		Page

Abstract

A key component of Air Force Civil Engineering project management is facility programming, the identification of requirements for construction projects. The literature review revealed that inadequate identification of facility requirements has lead to unsatisfied facility users, excessive cost growth, rework of construction documents, loss of projects, and change orders during construction.

The goal of ~~the research~~ was to identify potential improvements to the programming processes used by the Air Force. The "Delphi Technique" was used to solicit information about programming from two panels of "experts": (1) chief engineers within Base Civil Engineering organizations, and (2) professional programmers outside the Air Force. The respondents answered questions about programming in two rounds of questionnaires. Comparisons were made between the groups about current practices and attitudes about programming.

The research uncovered significant differences between how the two groups view and use facility programming. From the conclusions, the researcher proposed a new programming model that solves some current problems, and takes advantage of "good" programming practices. The key features are that programming and conceptual design are interactive processes, and the emphasis on functional programming.

AN ANALYSIS OF AIR FORCE FACILITY PROGRAMMING AND ITS EFFECT ON DESIGN AND CONSTRUCTION

I. Introduction

Chapter Overview

This chapter provides background on the Air Force's design and construction management process and identifies the research's general issue: facility programming and how it effects design and construction. In addition, the chapter includes the problem statement, research objectives, research questions, and scope and limitations.

General Issue

One of the major missions of Air Force Civil Engineering organizations is the construction, renovation, repair and maintenance of Air Force facilities. Civil engineering accomplishes this mission through the management of facility design and construction projects. A key component of project management is facility programming.

The Air Force often misses it's goal of quality and maintainable facilities on time and within budget. The current programming process has contributed to the problem. Facility programming identifies the functional and technical requirements for proposed construction projects involving buildings and infrastructure on Air Force installations.

Inadequate identification of facility requirements has lead to excessive cost growth, rework of construction documents, loss of projects, and change orders during construction. Another problem connected with poor programming is unsatisfied customers, the facility users. Facilities not meeting the users' needs ultimately effect their job performance. In other words, the above problems cost the Air Force both time and money.

Background

Air Force Civil Engineering is responsible "for planning, acquiring land, designing and constructing installation facilities for the Air Force (1:3)." Civil Engineering accomplishes these responsibilities through the Air Force's design and construction programs. AFR 89-1, Design and Construction Management, states that:

> The primary objective of design and construction management is to acquire quality facilities on time and within available resources. The facilities must be reliable and maintainable, meet prescribed environmental standards, and enhance user productivity and livability. (1:3)

The design and construction management process begins with the identification of a project. The Air Force defines a construction project as:

> A plan of work necessary to produce a complete and usable real property facility or a complete and usable improvement to an existing real property facility. (2:103)

A project is further defined as work accomplished at one time to include any new construction, repair or maintenance work done by contract. In the Air Force, a contracted project will go through three phases: (1) programming, (2) design, and (3) construction.

Programming. AFR 86-1, Programming Civil Engineering Resources, lists three major elements of programming:

1. Determining the facility requirements needed to accomplish the mission.

2. Evaluating existing assets and determining the most economical means of satisfying the requirements.

3. Acquiring any additional facilities that are needed or work that must be done on an existing facility.

Air Force programming involves two primary tasks: (1) selecting the appropriate funding avenues and (2) preparing the necessary programming documents (2:6.1).

The five key funding avenues available for facility projects are the:

1. Operation and Maintenance (O&M) program. O&M funds are used to accomplish in service and contract work on base facilities excluding housing and certain non-appropriated requirements.

2. Unspecified Minor Construction (P-341) program. P-341 funds are used for base minor construction requirements which exceed $200,000 and cannot be programmed in the MILCON because of urgent need.

3. Military Construction Program (MILCON). MILCON includes new work costing more than $1,000,000 on base

facilities or new work costing more than $200,000 and less than $1,000,000 which does not meet P-341 program criteria.

4. Non-Appropriated Fund (NAF) program. NAF resources are used to support projects for Morale, Welfare, and Recreation (MWR), Temporary Lodging Facility (TLF), Army and Air Force Exchange Service (AAFES) and Air Force Commissary Service (AFCOMS) using surcharge funds.

5. Military Family Housing (MFH) program. The MFH program includes new construction and renovation of family housing units, mobile home parks, related facilities (e.g. family housing offices and housing maintenance facilities), and other community support facilities (e.g. parking areas, utilities, and playgrounds).

Normally, selecting the appropriate fund source involves three variables: (1) funding approval level, (2) facility type, and (3) work classification (2:6.1-24.1).

The second task, preparing the necessary programming documents, requires Base civil engineering (BCE) personnel "to work closely with the user to accurately and clearly identify and express needs (2:6.1)." The BCE personnel translate the identified needs into proposed facility projects. The programming documents outline the project requirements. The Air Force uses three programming documents, as follows:

1. BCE Work Request (AF Form 332). The AF Form 332 is used to request and justify a proposed project at the lowest approval level, the Base Civil Engineer. The AF Form

332 is a one page form briefly outlining the project requirements.

2. Military Construction Project Data (DD Form 1391). The DD Form 1391 is used "to request and justify a construction need" for projects requiring higher funding authorization (e.g. MAJCOM and Congress) (3:2-7). This document is usually one to three pages including a preliminary cost estimate, the project requirements, and a description of the proposed construction.

3. Project Book (PB). The project book is used to "collect data, criteria, functional requirements, and cost target information required for the design process (4:2)." The DD Form 1391 and project book are not required for all construction projects. The requirement for these documents depends the funding program and major command policy.

The latest developments in Air Force programming methods is the Requirements and Management Plan (RAMP). The RAMP is part of a new MILCON execution process. The new concept eliminates the Project Book and, more of less, replaces it with a new requirement, called a Project Definition. The objectives of the Project Definition are to increase user involvement, identify all functional requirements, and develop a good floor plan (5:4).

Design. Design is the process of translating the functional and technical requirements of a project into the necessary working drawings and specifications. AFR 88-15, Criteria and Standards for Air Force Construction, stresses design excellence:

> Achievement of excellence in design shall be the primary goal for all construction projects. Reaching this goal requires a commitment by designers and administrators to architectural quality, which includes the relationship of architecture to the surrounding community as well as the details of design that effect the users of the building. (6:1-1)

Air Force design is either performed by in-house (e.g. BCE) personnel or by contracting services from an Architecture-Engineering (A-E) firm.

When using an A-E firm, the design is reviewed at various stages, usually at 35, 65, and 95 percent design completions. Air Force design managers are required to "review every project, regardless of program type, for technical and functional adequacy (1:5)." AFR 89-1, Design and Construction Management, defines functional and technical reviews, as follows:

> Functional Review. A review to include the user's requirements in the design. Project designers guide the user through the design to help the user to fully understand the drawings and specifications as they relate to their requirements.

> Technical Review. A review to verify the technical sufficiency of the design. Reviewers ensure functional adequacy, provision of technical requirements, adherence to Air Force criteria, and identify and remove design deficiencies before contract award. (1:28,30)

The designer/A-E formally submits the design at the review points to show design development. The design team then checks "for compliance with design criteria, maintainability and changed criteria, limit wasted effort on misdirected design and comment on design acceptability (3:3-10)." The design team members are representatives of the using agency, BCE, the MAJCOM, the design manager, the design agent and the designer/A-E.

Through the designers' efforts, project reviews and other communication the design progresses to completion. The final product includes working drawings, specifications, and cost estimates that become part of the contract documents. The design process ends with construction contract award (3:3-3).

Construction. The final phase of the project management is construction.

> The contract award marks the point at which the criteria, the needs, the concepts and ideas discussed in the course of design begin to become reality through the actual efforts of the construction contractor. (3:3-43)

During the construction process, Air Force construction managers monitor costs, schedule, quality and the effect they have on customer satisfaction.

Construction contract changes, or change orders, occur during construction. They fall into three categories: (1) mandatory changes, (2) optional changes, and (3) user changes.

> Mandatory Changes:
> 1. Actual conditions found on the construction site are not compatible with drawings and specifications.
> 2. Unknown or unforeseen conditions make change necessary.

> 3. Obvious technical errors or omissions in the drawings and specifications must be corrected to adequately define work.
>
> Optional Changes. Changes in basic design criteria since design was completed, omissions in drawings and specifications, contractor proposals, and other improvements in design.
>
> User Changes. Revised operational mission or equipment requires a change in the facility. (1:27)

Change orders have the most potential to effect the project by impacting cost, schedule and quality. The Air Force's management of modifications maybe it's most important role in the construction process.

The construction phase ends with facility acceptance. Air Force personnel conduct prefinal and final inspections to identify defects, and direct the contractor to correct defects. The Air Force accepts the completed project after final inspection acceptance. "This point marks the date that the facility is ready for occupancy by the user (3:4-29)."

Problem Statement.

Facility programming has significant impacts on the Air Force's design and construction goal "to satisfy the user's needs with quality construction (1:4)." However, current programming processes do not adequately define and communicate facility requirements in support of design and construction. Therefore, this research was examined the Air Force's current programming methods, identified alternative methods, and developed a new programming process to identify and translate user needs into quality facilities.

Research Objectives

To develop a new programming process to improve the design and construction of Air Force facilities, the following research objectives were identified:

1. Identify the weaknesses and strengths of the programming processes used by the Air Force.

2. Identify the weaknesses and strengths of the programming processes used by commercial Architect-Engineering firms.

3. Combine the successful elements into a new programming model.

4. Recommend ways to test and validate the new programming model.

Research Questions

To achieve the research objectives, the following investigative questions must be answered:

1. What effect does the programming process have on facility projects?

   a. How does programming interact with design?

   b. How does programming impact construction?

2. How does the Air Force agencies program facility projects?

3. How do commercial Architect-Engineering firms program facility projects?

4. What programming methods produce quality facilities?

a. What are the key elements in successful programming methods?

b. How can the key elements be identified?

5. How can the Air Force incorporate the key elements of successful programming into its own design and construction management process?

Justification for Research

Air Force Civil Engineering does not adequately program facility projects. Prior research has identified poor project definition as a major cause of problems in design and construction management. In addition, Civil Engineering personnel at Air Staff have initiated improvements to the MILCON program. Inadequate programming procedures are one reason for the proposed changes.

Dutcher, in his thesis, identifies inefficiencies in the Military Construction Program (MILCON). The research was based on the perceptions of personnel working at Air Force Regional Civil Engineer (AFRCE) field offices, the major commands (MAJCOMs), and Air Force bases. Inadequate definition of scope during programming and ineffective technical and functional reviews were the major problems. The following conclusions and recommendations were made:

1. The base's are not providing all the necessary information for project design.

2. The MAJCOMs need to ensure the project book provides better support to the Architect-Engineering firm.

3. Lack of time was a significant reason for project book inadequacies.

4. The project review process is excessively long causing delays.

5. Design errors cause delays and change orders during construction.

6. The bases need to improve definition of project requirements in DD Form 1391s and project books.

7. Design and construction are hindered by extra levels of management.

8. Base personnel are not adequately trained to identify user needs (7: 86-90).

Mogreen, in his thesis, identifies the causes and cost of changes to military construction contracts. The study reviewed 25 construction projects, administered by the Corps of Engineers, for reasons and costs of 778 changes contained in 268 modifications. The primary causes of the modifications were: (1) design deficiencies (36.3%), (2) user requested changes (22.3%), and (3) unknown site conditions (21.8%) (8:58). Inadequate programming was recognized as the main reason for user requested changes. Mogreen writes:

> In general, it appeared that poor project scope definition was a major contributor to user requested mods. Projects were designed and let for bid without a firm scope definition being communicated to the designer or user. Consequently, the designer may not have been aware of what the customer wanted and the customer not aware of what was designed until construction actually began. (8:82)

Stollbrink, in his thesis, studied user involvement in the Military Construction Program (MILCON). The analysis involved surveying using organizations for 104 MILCON projects completed in fiscal years 1984 and 1985. The research investigated (1) user involvement in the programming and design phases, and (2) the relationship between user involvement and changes in a MILCON project. He notes that:

> Changes during the programming phase usually do not pose major problems as they often involve changing the scope of the project. However, this could delay project approval and if the scope change is large and occurs after the project had been approved it could delay or kill the project.
>
> Changes during the design phase can cause more significant problems, especially if they require an increase in project scope and/or a major redesign effort. Changes during the construction phase are typically very expensive and should be avoided at all costs.
>
> Changes during the design and/or construction phases can also cause costly time delays. Changes during the design phase can also result in possible loss of the project due to increases cost. (9:1-2)

Stollbrink identifies project books containing insufficient or out-dated information as one possible cause of change requests during design and construction. The research suggested that the users may not have passed on all the necessary requirements to BCE personnel and that many users were not aware of the purpose of the project book. The results of the study indicated room for substantial improvement in facility programming with:

1. 38.1 percent of the users not aware of that the project book is the basis for project design.

2. 34.8 percent of the users not aware that providing functional requirements was an important part of their input to the project book.

3. 26.7 percent of the users indicating the project book did not adequately describe the projects functional requirements (9:27-31).

Another indication of problems with identifying project requirements are current changes to the MILCON process with the objective of improving execution, increasing quality, and reducing costs by redefining programming and design. The current process is lengthy and expensive with changes difficult to predict and facility quality suffering because project requirements are frequently unidentified. The center of the new procedure is the Requirements and Management Plan and the Project Definition. The Project Definition is a type of programming document. Its main objective is to increase user involvement in defining functional requirements. The RAMP incorporates planning information and project requirements into a guidance package for the design and management team. The procedure encourages the hiring a professional (A-E firm) to accomplish the Project Definition with emphasis on user involvement and identification of all building components. The expected results of the new project development procedure are a simplified process that costs less and improves quality (5:4-5;10).

The researcher, also, chose to examine facility programming for two other reasons. First, in the programming stage, effective decision making can have a positive impact on the total project costs. Figure 1 indicates that "the prospects for implementing changes are greater" without "the possible negative effects on project costs and construction schedules" in the beginning stages of project development (11:4). Second, facility programming is a growing architectural service outside the Air Force, because of its perceived benefits in producing better, quality buildings. An American Institute of Architects (AIA) study said that "the development of thorough programming techniques holds promise of being the most significant development in architecture in current times (12:93)."

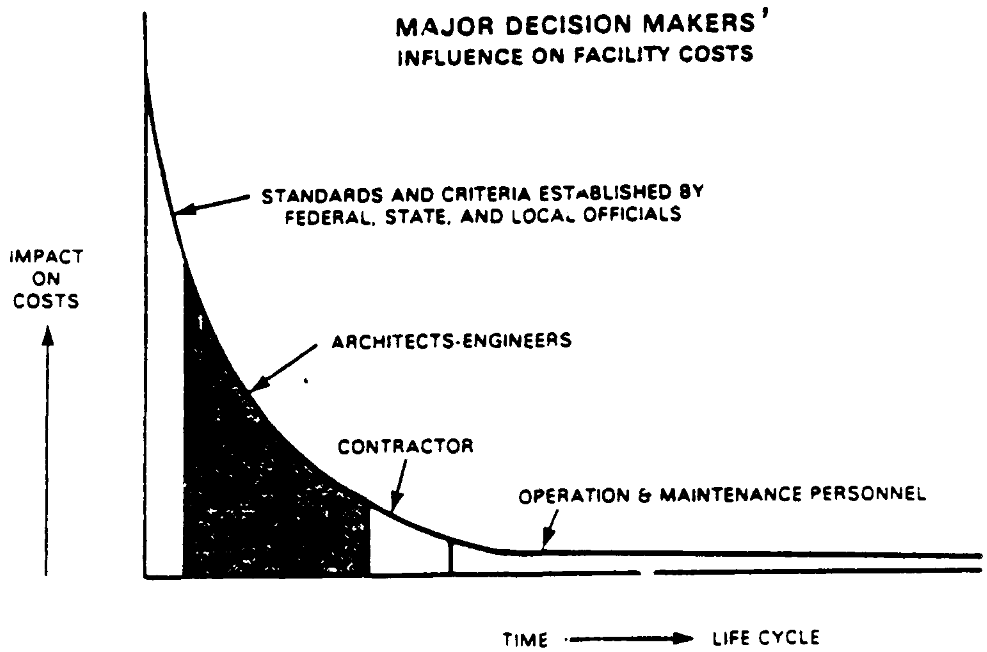

Figure 1. Major Decision Makers's Influence On Facility Costs (11:5)

Scope and Limitations

The research addresses facility programming and how it interacts with design and how it effects construction. Even though prior research has primarily studied the MILCON program, this research will also examine programming procedures for the (1) Operations and Maintenance (O&M) program, and (2) Non-Appropriated Fund (NAF) program. The research will also examine programming procedures outside the Air Force.

The research will include the following limitations:

1. The research will only address the programming of buildings, not infrastructure items, such as runways and utilities.

2. The research will only involve study of programming methods in the continental United States (CONUS).

Chapter Summary

A key component of the Air Force design and construction management is facility programming. However, the current programming procedures have contributed to problems such as excessive cost growth, rework of construction documents, and change orders during construction. Another problem associated with poor programming is unsatisfied customers.

A facility project has three phases: (1) programming, (2) design, and (3) construction. Programming determines the facility requirements needed to accomplish the mission. Design translates these requirements into working drawings and specifications. Construction is the actual building of the facility from the contract documents based on the requirements.

The purpose of the research is the development of a new programming process that adequately identifies the project requirements. The research will accomplish this goal through examination of programming methods used by the Air Force and commercial Architect-Engineering firms. By identifying the successful elements of several programming processes, the researcher can develop an improved programming model that meets the supports the goals of Air Force design and construction.

This chapter examined the current Air Force design and construction management process, including facility programming. The next chapter, the literature review, concentrates on programming procedures and methods outside the Air Force. The review looks at whose involved in programming, along with programming models and programming techniques.

II. Literature Review

Chapter Overview

The literature review covers several areas of interest in facility programming. The main topic areas included are: (1) participants in programming, (2) programming's interaction with design, (3) programming models, and (4) programming techniques. The literature aided in identifying areas of controversy and agreement within the field. In addition, the review mainly focuses on programming methods outside the Air Force. Air Force policy and procedures were summarized under Background, in the previous chapter.

General Information

Facility programming is not new. In fact, formal programming is traced as far back as 1862 to an architectural competition for new court buildings in London (13:4). In addition, very complete programs were a part of the Beux-Arts architectural education system (14:204). However, formal recognition of programming, as a distinct service, is fairly recent. One of the landmark articles on the subject is Pena and Caudill's "Architectural Analysis - Prelude to Good Design", first published in Architectural Record, May 1959 (15). Further, the "integration of the art and science of programming seemed to have reached its heyday by the early '70s, at least in the literature and in the press (14:203)."

The research in the programming field has consisted mostly of case studies and interview with professional programmers. One of the most extensive research efforts is White's Interviews With Architects About Facility Programming published in 1982. White interviewed 73 participants primarily about architectural programming education (16). However, the researcher's main source of information was Mickey A. Palmer's book, The Architect's Guide to Facility Programming. The book, itself, is an excellent literature review on programming. Palmer includes: (1) interviews with prominent programmers, (2) synopses of the major books on programming, (3) reviews of different programming models, and (4) programming case studies. In addition, a considerable portion of the book is devoted to the overview of 70 programming techniques. Though published in 1981, it is still one of most comprehensive books on facility programming (17).

Definition of Facility Programming

The researcher, in starting his literature review, searched for a commonly accepted definition of facility programming. However, the words "programming" and "program" have different meanings to a number of different groups. These groups include the U.S. military, architects, engineers and other professionals. For example, the sequencing of coded instructions for a computer is "computer programming." On the other hand, DOD has "major force programs" that acquire military resources or assets. These examples have little relation to facility programs or programming.

The term "facility programming" is used by individuals involved with the design of construction projects. But even this term is not universally applied within the design professions. Other common phrases are architectural programming, functional programming, design programming, space programming, and operational programming. The number of terms mirrors the many definitions found in the literature. (17:4-5)

However, the definitions for facility programming do have many common elements. First, programming is a systematic process of identifying the requirements for a facility project. Second, the process includes the collecting, analyzing, organizing, evaluating, and communicating of relevant information for facility design. Third, the programming information includes the client's organizational needs, goals and objectives. Fourth, the programming process produces a "program", usually in the form of a written and diagrammatic document. (17:3; 18:15; 19:xii)

Problem Identification. Another popular view of programming is that it is a problem-seeking process. In other words, programming identifies the problem that the design must solve. Therefore, programming and design are parts of a "problem cycle" which includes the problem identification and the problem solution, respectively. Figure 2 shows the parallels between a project development and problem solving. (17:7; 20:295; 21:14-15)

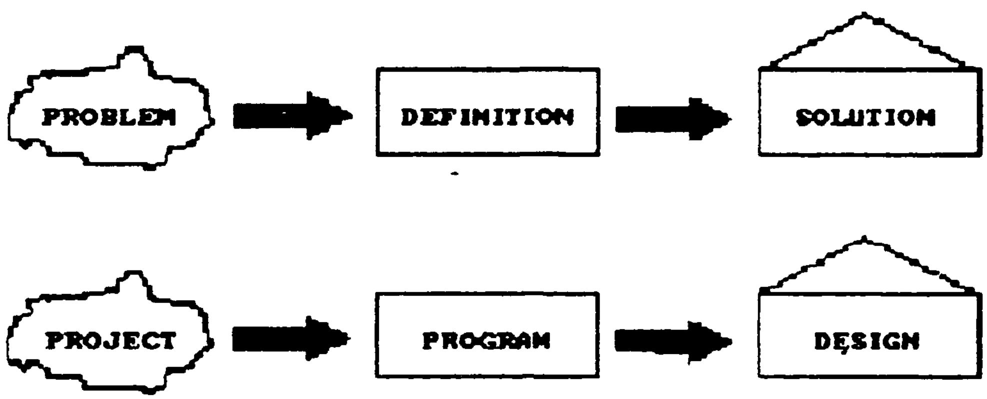

Figure 2. Parallels Between Project Development and Problem Solving (17:7)

__Types of Programs__. As mentioned above, one of the outputs of programming is the "program." However, many types of programs exist addressing specific needs and information requirements. Some examples include space programs, functional programs, and furnishings programs to name just a few. The program type depends on the project and the information needs of the client and designer. For facility projects, comprehensive programs are usually the best, because they address the total facility. The comprehensive program includes: (1) a master program, (2) a facility program, and (3) a component program. The master program is most useful to the client.

> A master program presents the most general type of information, providing an overview of all the significant design issues and summarizing the principal programmatic conclusions. (17:23)

However, the designer is generally more interested in the facility and component programs. The facility program provides the information base for design and for evaluation. The component program contains specific information on the engineering systems or the individual spaces within the facility. (17:22-23,46)

Participants in Programming

A discussion of the participants and their roles in programming is essential to understanding the process. Many individuals and groups contribute time, information and expertise to the programming process. The Architect's Guide to Facility Programming names three main categories of participants: (1) the programmer, (2) the client, and (3) the designer (17:10). However, the Air Force programming process does not always follow the traditional professional relationships found outside governmental work. Therefore, an expanded look at the participants and their roles is needed.

Programmers. "The programmer is the firm or individual who conducts the programming and produces the program (17:11)." The client, the project architect, or outside consultant can fill the role of the programmer. In the traditional client-architect relationship, the client is responsible for providing the program. Programming may be performed by the client's in-house planning or programming staff. They may also hire an outside consultant (other than the design firm) to produce the program.

However, clients' have increasingly delegated the programming responsibility to the architect or designer. Reasons for this shift in roles include increasingly complex and specialized buildings, and lack of client programming expertise. In practice, the design firm, is at least responsible for reviewing cr verifying the programming information. (17:9,14)

As mentioned before, an outside or third-party consultant can also perform programming. These may be planners, engineers, space management consultants, interior designers, and other professionals. Often, another architect, other than the designer, is hired to program a facility. (17:14)

In the Air Force, programming is usually performed by the Civil Engineering in-house programming staff. Air Force programmers are military or civilian architects, engineers or planners. However, the new MILCON RAMP process is advocating hiring Architect-Engineering firms to perform many of the programming functions. (2:6.1; 10)

Clients. The term "client" refers to the individual or organization that employs the programmer and designer for a facility project. The client often is the owner and user of the facility, but not in all cases. The client plays a major role in the programming process. First, clients are a primary source of programming information. Second, one of the principal purposes of programming is the identification of the client's needs and requirements. Third, in successful programming, the client must ultimately buy-in or approve the program. (17:20)

Civil Engineering provides programming and design services with both in-house personnel and by hiring Architect-Engineering firms. When performing work in-house, the user or using agency becomes Civil Engineering's "client" or customer. However, when using A-E firms, Civil Engineering fills the role of client. (6:1-1)

Owners. The term "owner" is often used synonymously with client. However, often they are not the same. The future owner commissions a building for several reasons. Two main reasons are: (1) to provide the owner or owner's organization a suitable operating environment for living, working or some other use, and (2) to provide an economic return on the owner's investment. (22:6-7)

Like clients, owners are important in the programming for the many of the same reasons. Owners and clients are both recipients of programming information. They are interested in

> data that enables them to judge the worth of a project: costs of designing, constructing and operating; future use, functional efficiency; the amount of time it will take to build and occupy the facility. (17:17)

On a typical Air Force base, the base or wing commander may fill the role of "owner." Facilities are the commander's resources or assets to that enable him to fulfill the base's mission. As such, the commander's inputs during facility planning and programming are essential.

Users. The facility users are, also, important in programming because "it is the user who ultimately interprets the design (14:208)." In addition, there is a difference between the client-owner and the facility's "users." The client or owner may not actually occupy or actively use a new facility. In addition, even if the client is a "user", the client could be one of many individuals or groups benefiting from the proposed construction. In other words, the client's and users' needs may be different for a proposed project. The users, then, are another primary source of functional information in the programming process. (14:204; 23:3)

In addition, users fall into different categories: (1) facility occupants, (2) the occupants' clients, (3) facility managers or operators, and (4) the general public. These groups may all have important, but distinct, inputs to the facility's operation. For example, the general public, represented by the local community, is affected by the location and physical appearance of a new building. (17:10)

Designers. The term "designer", usually "identifies the architect-of-record (17:11)." However, in the Air Force, the designer often is an engineer. In addition, when an architects or engineers supervise Air Force A-E contracts, they are called "project managers." Project managers are mentioned because they have many of the same concerns and roles as designers in the programming process (3:1-5). The designer's part in programming is important because "the firm or individual is the principal user of the program and interprets it in the development of design (17:11)."

Architects. As previously stated, architects fill the roles of both programmer and designer. In programming, architects are important sources of information. They have expertise in building construction and often with particular facility types. Also, often architects are familiar with programming methods and techniques. (17:14; 23:3,8; 24:2)

Though programming is "not exclusively the architect's domain (17:14)", architects represent one of the principal professions that provide these services. Programming is recognized as a predesign service and a separate discipline within the architecture profession. Evidence of programming as an established architectural service include: (1) architects specializing in programming, (2) architectural registration exams including questions about programming, and (3) architecture schools providing courses on programming. (22:10; 24:2)

As designers, "the architect serves as creator, coordinator, and communicator of the project's design in overall concept and in all of it's parts (22:8)." As creator, the architect is responsible for producing the functional, aesthetic, and technical design for a facility. The architect, also, coordinates the project by directing the work of other design professionals, chiefly the engineering disciplines. The architect also acts as the client's representative in the construction of a facility. Finally, as a communicator, the architect explain and justify the design to all the parties involved.

Engineers. Though engineers, normally, are not the programmers nor the lead designers on building projects, they deserve special consideration in this study. First, in Air Force Civil Engineering, engineers often fill roles, such as programmer and designer, that are usually the architect's domain in private practice. Second, architects use engineers as consultants in both programming and design. Third, the engineers are the lead designers on projects where the engineering work dominates (for example, runway construction). (22:8)

Engineers have specialized knowledge in their areas of expertise valuable in both programming and design. In building projects:

> approximately 25 to 50 percent of the construction cost may be embodied in the structural, mechanical (that is, plumbing, fire protection, heating, ventilation, and air conditioning), and electrical systems. (22:8)

Though programming is more often concerned with functional needs, it may also include specialized information on the technical building components.

Consultants. As mentioned previously, architects are not the only professionals who specialize in programming. Often, planners, industrial designers, interior designers, management consultants, and behavioral scientists are professional programmers. The client may hire these professionals directly to program facilities or the architect may employ them as consultants. (17:9,14,42)

Benefits of Programming

Programming provides benefits primarily for two groups: (1) the clients, and (2) the designers. The literature states many advantages in the programming of facility projects. As a mechanism to collect, analyze, organize, evaluate, and communicate information, programming can provide financial and organizational benefits for both the clients and designers.

Programming is described as a systematic and analytical process. As a systematic process, it ensures that all the important and relevant project issues are addressed. As an analytical process, it allows the client and designer to make decisions based on factual information. (17:3)

Further, programming aids the client to: (1) define organizational requirements, (2) identify operational improvements, (3) document organization or operational structures, and (4) plan future organizational change and growth. In turn, the designer can better understand the client's operation and design a facility that responds to the client's present and future needs. As a result, maximum operational efficiency and productivity is provided within project constraints (i.e. budget). (12:93; 18:5,37; 25:45)

Programming also provides benefits as a communicative tool. Many programmers feel clients can contribute most to a project's success during programming. Programming provides the clients and users the opportunity to communicate their requirements prior to design. First, programming can be an effective vehicle for soliciting active client and user participation. Second, feedback is a crucial element in the programming. Feedback allows the client and users to evaluate whether their needs are clearly stated and understood. Third, programming encourages the active client and user involvement that is essential for securing commitment to the program. Fourth, the programming process is a format for resolution of differences between client and designer, or between user groups. Finally, all this provides a framework for effective interaction between the client and designer. (13:4; 14:204-206; 18:82; 23:8)

As an evaluative tool, programming tests design proposals and alternatives avoiding wasted time on irrelevant solutions. For the designer, this provides more time for meaningful design. For the client, this equates to eluding a compromise design brought on by time pressures. Another benefit is early design satisfaction that can reduce the overall time to complete design work. (12:93; 19:2)

Many of the advantages of programming result in monetary savings. First, programming clarifies the scope of design work providing a framework for fair compensation. Second, programming reduces the possibility for omissions or errors that can cause expensive changes during design and construction. Third, programming examines project feasibility, possibly avoiding design and construction work that is not required or is not within funding limitations. Fourth, programming can address long term costs in the form of (1) energy consumption, (2) maintenance costs, (3) life-cycle costs. (17:3; 19:2)

In addition, the designer benefits from quality programming by increasing his profit margin. An American Institute of Architects (AIA) study, "Economics of Architectural Practice" linked program quality to profitability. Programs which were rated as "good" pretax income averaged 11.8 percent. On the other hand, for "poor" programs the average was only 7.9 percent, a reduction of 33 percent. In addition, programming costs are a small part of the total building costs. A survey revealed programming was only 0.25 to 0.50 percent of the construction costs. In other words, the client investment is trivial compared to the potential benefits to himself and the designer. (12:94; 18:18; 26:32)

Programming and Design.

One of the primary purposes of facility programming is to describe the project requirements for the design phase. The factors that influence design are addresses in some fashion during the programming process. Palmer classifies information under three main categories: (1) human factors, (2) physical factors, and (3) external factors (Figure 3). In addition, programming develops performance statements to guide design. (17:19; 23:2)

Further, when talking about the programming process, one must, also, address the design process. Programming and design are described as interdependent. They are closely linked and both part of the larger process that produces construction projects. However, the programming - design relationship is one of the most controversial subjects in the programming field. The debate is over whether programming is part of the design process or a distinct separate function. How programming interacts with design characterizes the philosophy behind many programming methods. (14:207-208; 17:25-26)

The Architect's Guide to Facility Programming describes the three main approaches to programming as (1) segregated, (2) integrated, and (3) interactive (Figure 4). Individuals using the segregated approach say programming is the initial step in the design process. However, programming is a distinct activity from design that requires different skills and capabilities. In other words, different individuals or

Human Factors

Activities
Behavior
Objectives/Goals
Organization
—Hierarchy
—Groups
—Positions
—Classifications
—Leadership
Characteristics (Demographics)
Social Forces
Political Forces
Interactions
—Communication
—Relationships
—Transfer of materials
Policies/Codes
Attitudes/Values
Customs/Beliefs
Perceptions
Preferences
Qualities
—Comfort
—Productivity
—Efficiency
—Security
—Safety
—Access
—Privacy
—Territory
—Control
—Convenience

Physical Factors

Location
—Region
—Locality
—Community
—Vicinity
Site Conditions
Building/Facility
Envelope
Structure
Systems
—Engineering
—Communications
—Lighting
—Security
Space
—Types
—Dimensions
—Relationships
Equipment/Furnishings
Materials/Finishes
Support Services
—Storage
—Parking
—Access
—Waste removal
—Utilities (water, sewage, telephone)
Uses
Functions
Behavior/Activity Settings
Operations
Circulation
Environment
—Comfort
—Visual
—Acoustical
Energy Use/Conservation
Durability/Flexibility

External Factors

Legal Restrictions (Codes/Standards/Regulations)
—Building
—Land use
—Systems
—Energy
—Environment
—Materials
—Safety
—Solar access
Topography
Climate
Ecology
Resource Availability
Energy Supplies/Prices
—Conventional
—Solar
—Alternatives
Economy
Financing
Time
—Schedule
—Deadlines
—Operations
Costs/Budget
—Construction
—Materials
—Services
—Operations
Costs/Benefits

Figure 3. Factors Influencing Facility Design (17:19; 23:2)

groups should perform each function. Another characteristic of segregated methods is that programming is accomplished prior to design. Similarly, programmers using integrated methods see programming as the integral first part of design. The difference, though, is that "programming is design" not a predesign service. "The implication is that an architect [designer] must program and that a programmer

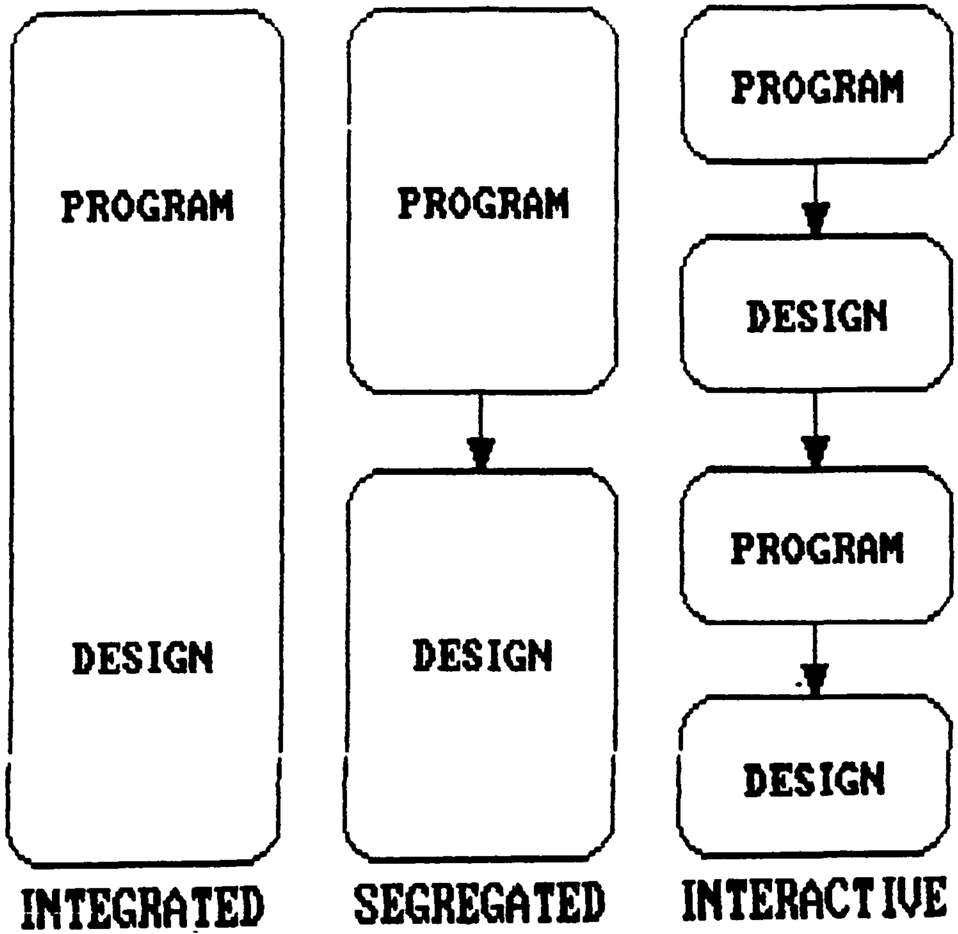

Figure 4. Three Approaches to the Programming/Design Process (17:28)

should be an architect [designer] (17:26)." With interactive methods, a project starts with programming then moves to design in iterative cycles. In other words, "the program and design are developed in alternating sequences and in response to each other (17:26-27)." The cycle includes (1) programming, (2) design , and (3) evaluation and review. The cycle repeats itself until the design process is completed. (17:25-27)

Programming Methods

In reviewing the literature, there are many distinct programming methodologies. These methods were developed by architects and other programming professionals. The Architect's Guide to Facility Programming, though, describes three common characteristics most programming methods exhibit. They are that the programming processes are (1) systematic, (2) iterative, and (3) progressive. (17:24)

Programming methods are systematic because they follow certain procedures. This allows the programmer to rapidly, accurately, reliably, and economically gather and present the needed programming information. The programming process is also described as iterative (Figure 5). Many projects involve large amounts of data. This information is usually accumulated through "iterations" or cycles. The programmer

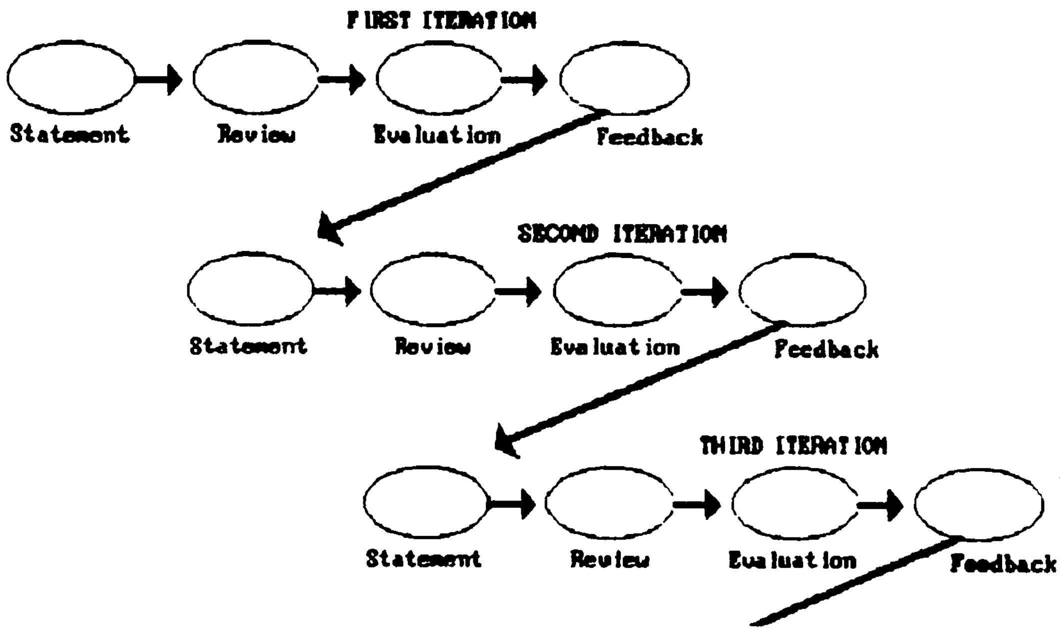

Figure 5. Iterative Programming Cycle (17:27)

and other programming participants expand and refine data through periodic reviews and evaluations. Closely related to the iterative process, is the idea that programming is progressive. To make the information more manageable, programmers first collect general information. Then they build on the information working towards specific programming requirements. (17:24-25)

Pena Model. One of the most popular and enduring programming methodologies was developed by William Pena of the firm CRSS. In a recent article on programming in Architecture is was described as a "good base line model (14:207)." Pena explains the model in his book, Problem Seeking, now in its third edition (21).

Pena advocates giving the designer the programming information in two stages working from general information to specific requirements. The two phases are the schematic program and program development "related to the two phases of design - schematic design and design development (21:40)." Figure 6 illustrates this programming - design relationship.

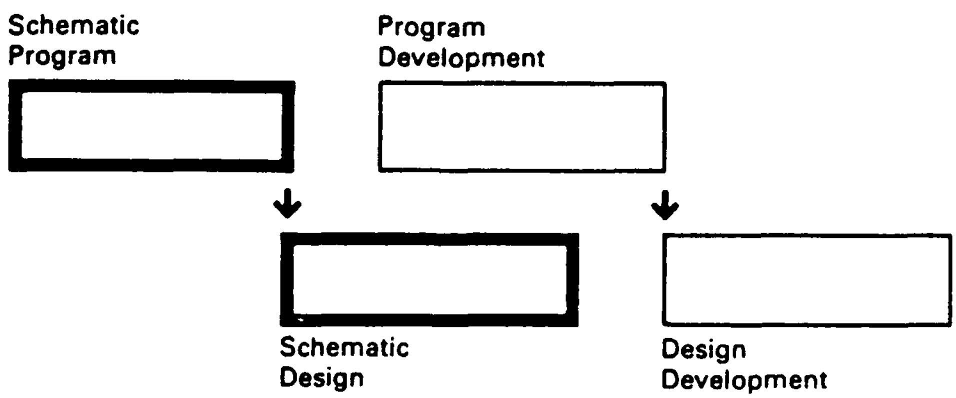

Figure 6. Two-Phase Programming Process (21:40)

The Pena Model is a good example of the segregated approach to programming. Programming and design are separate disciplines, with different functions. Programming is described as analysis, the process of identifying the problem. On the other hand, design is synthesis, or problem solving. (21:18-19)

The core of the programming method is the use of an information matrix. One side of the matrix are five procedural steps, as follows: (1) establish goals, (2) collect and analyze facts, (3) uncover and test concepts, (4) determine needs, and (5) state the problem. The other side is composed on four informational components: (1) form, (2) function, (3) economy, and (4) time (Figure 7). The components are "big baskets" to collect the programming data instead of having a large number of information categories. This concept simplifies the process. The importance is placed on putting the information somewhere and avoiding duplication. (14:207; 21:12-13)

Function	1 people 2 activities 3 relationships
Form	4 site 5 environment 6 quality
Economy	7 initial budget 8 operating costs 9 life cycle costs
Time	10 past 11 present 12 future

Figure 7. Four Considerations in Programming (21:30)

Davis Model. Gerald Davis and his firm TEAG (The Environmental Analysis Group Ltd.), located in Ottawa, Ontario, Canada, "specializes in prearchitectural programming (17:29)." They view programming as including two distinct activities with three types of programs. The three program types are: (1) the functional program, (2) the technical program, and (3) the design program. The first activity includes the functional and technical programs, while the second activity includes the design program. (17:29)

Further, the two activities are "performed separately and by separate teams (17:29)." The first two programs are prepared by the client or his consultants. The third program is the responsibility of the designer. The Davis Model also includes a list of predesign activities that the client or programming consultant might perform to develop the functional and technical programs. (17:29)

Farbstein Model. Jay Farbstein of Jay Farbstein and Associates has used a five step programming method. The steps include: (1) literature survey, (2) user description, (3) performance criteria, (4) program options and costs, and (5) space specifications (Figure 8). "Each phase contains specific tasks and data considerations (17:33)" for either the programmer and client. The Farbstein Model stresses the involvement of both the owners and users with user needs a major consideration. (17:33-34)

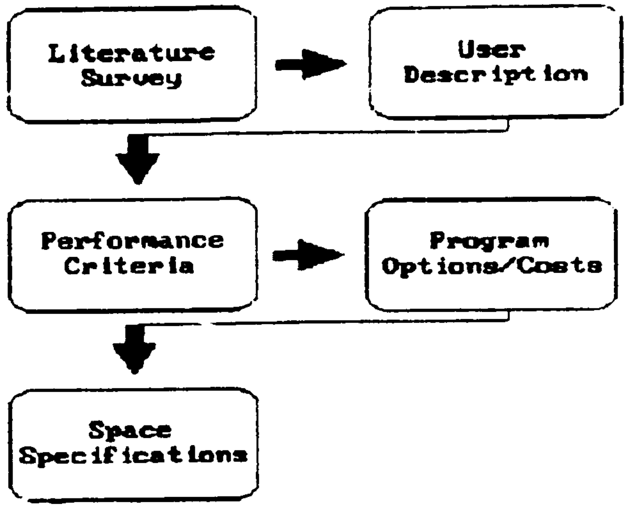

Figure 8. Farbstein Programming Model (17:33)

<u>Kurtz Model</u>. The Kurtz Model is a good example of the interactive or iterative approach to programming. John Kurtz developed a programming method which has four phases: (1) orientation, (2) base program, (3) iterative programming, and (4) design-as-feedback (Figure 9). Kurtz describes the method as hierarchical and sequential moving from general to more detailed requirements. Further, the programming occurs simultaneously and interactively with design, construction and occupancy (17:36). Only general programming decisions are made prior to starting design. After the base program is established, "successive iterations of program and design respond to each other and are revised accordingly (17:36)." The philosophy behind the programming method is that "users and needs will change continuously, therefore requiring continuous reprogramming (17:36)."

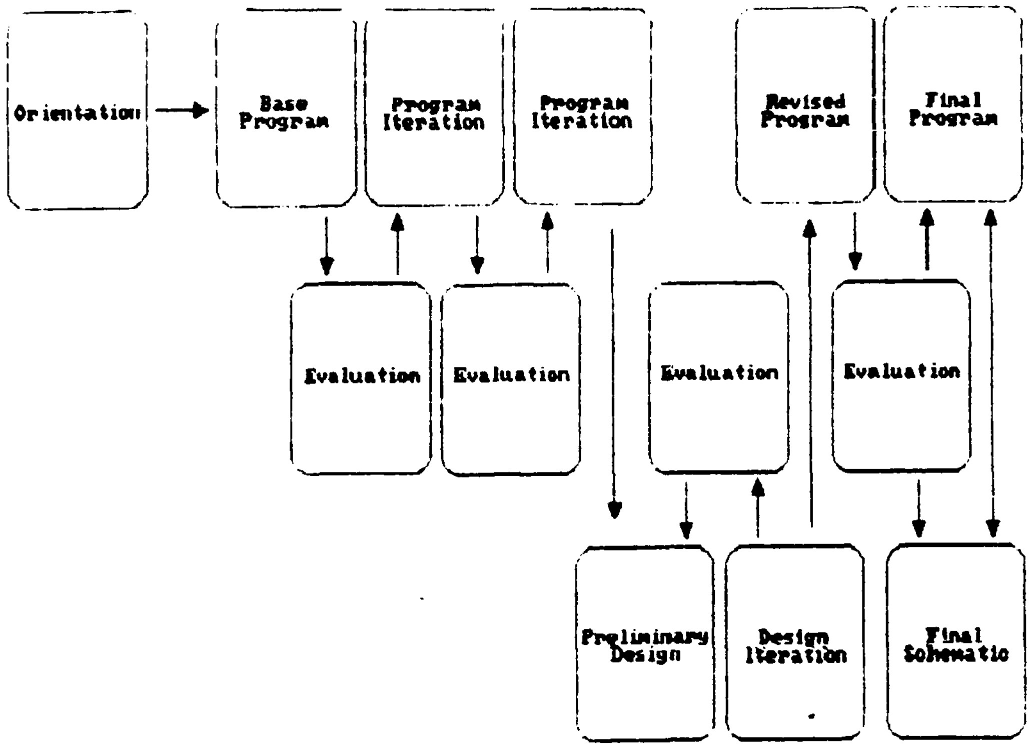

Figure 9. Kurtz Programming Model (17:35)

__White Model__. Edward T. White III stresses, what he calls, "a view of design" as a major influence in the programming - design model (18:5). Whites says programming is part of the design process. Further, programming is part of the view of design sequence that includes nine events:

1. Reality (laws, principles).
2. Search for and discovery of laws and principles (fact-making).
3. Known facts.
4. Gathering of facts.
5. Analysis, evaluation and organization of facts into meaningful patterns.
6. Response to facts in design synthesis.
7. Building product.
8. Building consequences.
9. Evaluation. (18:7)

White believes that a designer's personal attitude and values make up his "view of design." A programmer must share or understand the designer's view of design in order to provide "a smooth transition from problem statement to solution (18:5)." If not, the designer may find the program difficult to use.

White, also, stresses maximum interface between programming and synthesis (design). The programming and design processes should be continuous. He says, "the stronger the distinction between programming and design, the greater the chances that the spirit of the program will be lost (18:74)."

From White's book, Introduction to Architectural Programming, "the process of programming is composed basically of gathering, analyzing, evaluating, organizing and presenting information pertinent to the design problem (18:15)." In addition, the program format includes four types of data: (1) goals, (2) facts, (3) precepts, and (4) concepts. Taken from programs produced by White, his programming methodology includes tasks divided into three phases: (1) preprogramming, (2) programming, and (3) postprogramming. "The actual investigation or research work is what he calls programming (17:41)." (17:40-41; 18:16)

Wade Model. John W. Wade in his book, Architecture, Problems, and Purposes, describes three stages in the design process: (1) programming, (2) planning, and (3) design (also the term for the entire process) (Figure 10). The process

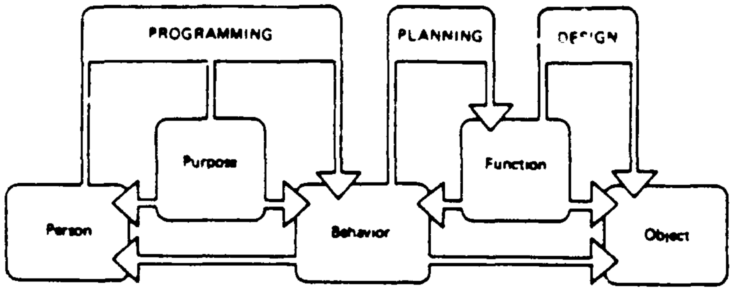

Figure 10. Design Phases and the Information Spectrum (20:84)

begins with programming. Programming is primarily "the collection and organization of information that is required for building design (20:83)." Unlike most other programming - design models, planning is the link between programming and design. Wade says planning convert programming information into "visual diagrammatic notation" (i.e. a bubble diagram). Simply, planning "diagrams building functions (20:83)." In the last stage, design, the designer develops the details, drawings and specifications for building construction. Wade describes the entire process in terms of "transformations of information (20:83)."

> Programming collects information about the person (client) and his purposes and converts it into information about behaviors (activities); planning takes information about behaviors and converts it into information about functions; design takes information about functions and converts it into information about objects (the building). (20:83)

The Wade programming methodology is illustrated as a flow chart with 15 possible steps (Figure 11). However, the most important steps are: (1) beginning a program, (2) developing a program, (14) preparing the program, and (15) presenting the program (27:191-194). Wade examines these

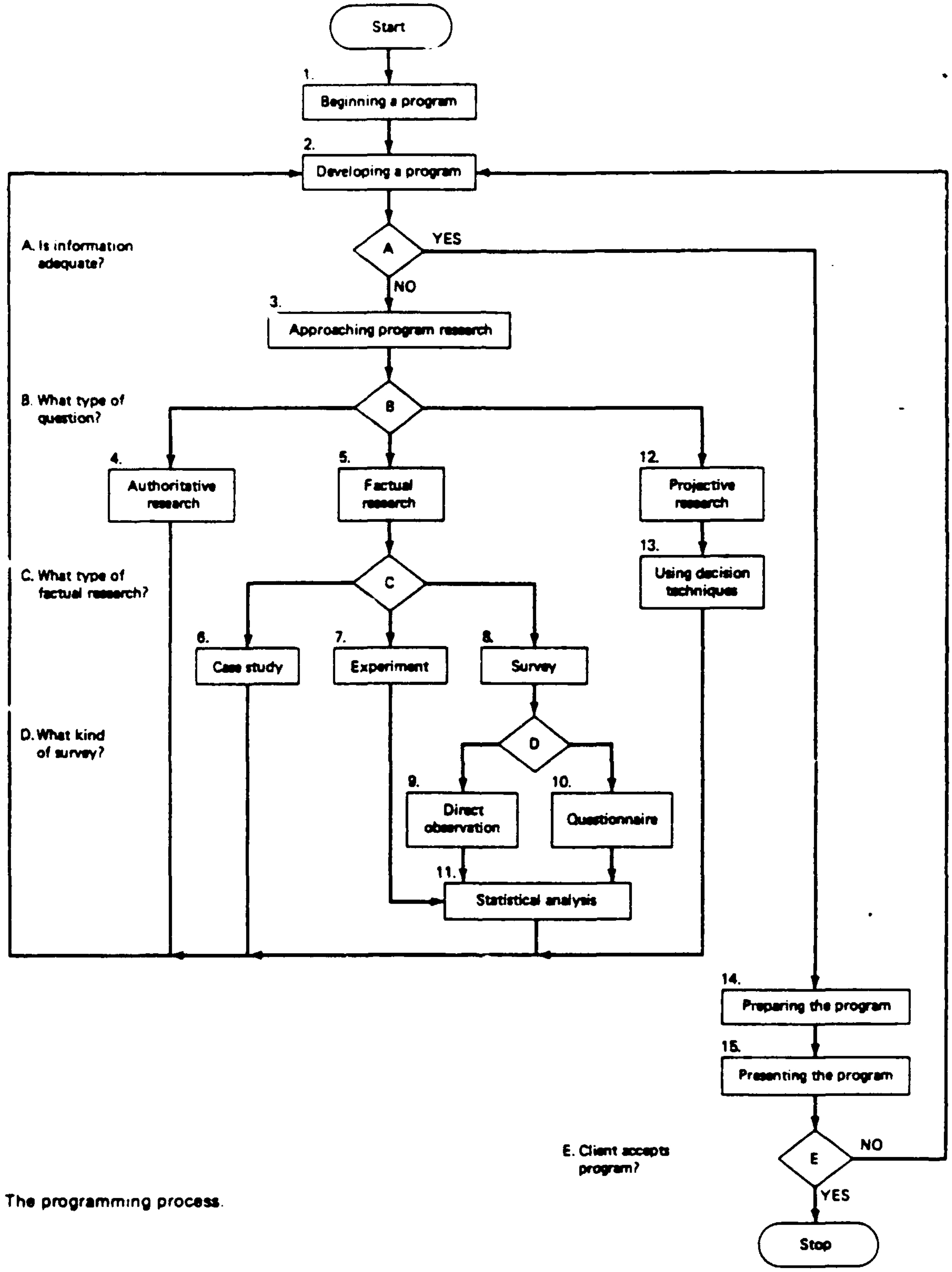

Figure 11. Wade Programming Model (27:193)

four steps in more detail because "they must all be done regardless of the simplicity or complexity of the program (27:192)." The Wade Model, also, uses four questions (or decision points) to determine if the other steps are necessary for more complicated facility projects. The questions are:

1. Is the information adequate?
2. What kind of information is needed?
3. What type of factual research is required?
4. What kind of survey should be used? (27:194)

Kemper Model. Alfred M. Kemper, the author of the Architectural Handbook, describes a programming process similar to the Pena Model. The Kemper programming methodology includes two stages: (1) a schematic or conceptual program, graphically expressed in the schematic design, and (2) a more detailed program, leading to design development (25:182). Both programming stages follow five procedural steps. The steps are:

1. Definition of Client's Objectives.
2. Collection, Organization and Analysis of Facts.
3. Evaluation of Alternative Concepts.
4. Determination of Space Requirements.
5. Statement of the Problem. (25:182,184)

Kemper uses a format outline called a "program guide." The program guide helps "owners/users express their concepts (25:184)" in three general categories. The information is classified as either (1) goals and objectives, (2) functional needs, or (3) basic space requirements. (25:184)

The Kemper Model works by first identifying general information and moving towards more detailed requirements.

Kemper expresses this idea with four levels of programming information. First, a master plan for the "total complex" is developed. Second, a "total building" program is written for the each building in the project. Next, program units called "activity centers", groupings of related functions or spaces, are developed. Last, information on "individuals spaces" is detailed. (25:184-185)

Programming Techniques.

Programmers use a wide variety of techniques to handle the specialized information needs for construction projects. Programming techniques are closely linked to programming methods. In fact, these techniques are often called "methods." For the purposes of this research, a "method" refers to an entire programming model or process, while "technique" defines a procedure to manage a specific type of programming data.

There are large number of programming techniques. Palmer in The Architect's Guide to Facility Programming reviews 70 different techniques used in facility programming. Programming techniques are used to collect, analyze, organize, communicate and evaluate data (17:49). Figure 12, shown on the next two pages, lists these techniques by their primary and secondary purposes.

Another good source of information on programming techniques is Henry Sanoff's Methods of Architectural Programming. He reviews approximately 30 techniques listed

TECHNIQUE	Collection	Analysis	Organization	Communication	Evaluation
Background Data Research	■	□	■		
Surveys	■	□			□
Interviews	■				□
Questionnaires	■				□
Data Logs	■		□	□	
Standardized Data Forms	■		■	□	
Direct Observation	■	□	□		
Tracking	■	□			
Participant Observation	■			□	
Behavior Mapping	■	□	□	■	
Behavior Specimen Record	■	□			
Instrumented Observation	■			■	
Semantic Differential	■	■			□
Adjective Checklist	■	■			□
Attribute Discrimination Scale	■	■			□
Ranking Chart	■	■		□	□
Preference Matrix	■	■		□	□
Descriptive Statistics	□	■		□	
Inferential Statistics		■			■
Behavior Setting Survey	■	■	□		■
Activity Site Model	■	■	■	□	■
Time Budget Analysis	■	■	□		
Pattern Language	□	□	■	■	□
Space Unit Standards	■	■	□	□	
Space Program		□	■	■	
Energy Budgeting	□	■			□
Project Cost Estimating	□	■	■		
Construction Cost Estimating	□	■	■		
Life Cycle Cost Analysis		■			■
Value Analysis		■			■
Cost-benefit Analysis		■			■
Bar Chart/Milestone Chart		■	■	□	□
Activity Time Chart	□	■	■		
Critical Path Method (CPM)		■	■	□	■
Program Evaluation & Review Technique (PERT)		■	■	□	■
Precedence Diagramming Method (PDM)		■	■	□	■

Figure 12. Programming Techniques and their Information Processing Functions (17:50-52; 23:6-7)

TECHNIQUE	Collection	Analysis	Organization	Communication	Evaluation
Relationship Matrices	□	■	■	■	□
Social Map		■	■		
Sociogram		■	□		
Behavior Map	■	□	□	■	
Bubble Diagram		□	■	■	
Link-Node Diagram		□	■	□	
Block Diagram			■	■	
Interaction Net		■	■	□	
Dual Graph		■	■	□	
Adjacency Diagram	□	□	■	■	
Functional Relationship Diagram	□	□	■	■	
Layout Diagram			■	■	
Flow Diagram		■	■	□	
Organizational Chart	□	□	■	■	
Analysis Cards	■	■	■	■	□
Worksheets	■	■	■	□	□
Brainstorming	■	□		■	
Synectics	□	■		■	
Buzz/Rap Session	□			■	
Role Playing		■		■	□
Gaming	□	□		■	■
Group Planning	□	□	■	■	□
Narrative				■	
Graphics			□	■	
Audio/Visual Aids			□	■	
Oral Presentations				■	
Forums	■			■	
Panel Discussions	□			■	
Work/Charrette/Primer Books	■			■	
Rating and Rating Scales		□			■
Ladder Scale	□	□			■
Rating Chart	□	□			■
Evaluation Matrix	□	□			■
Weighting		■			■

Key

■ = Primary use

□ = Secondary use

Figure 12. Programming Techniques and their Information Processing Functions (Continued)

in "the programmer' kit of parts" (Figure 13). The techniques are used for (1) problem identification and exploration, (2) searching for and expanding ideas, (3) classifying and analyzing ideas, (4) generating and evaluation alternatives, and (5) post occupancy evaluation.

	Problem identification and exploration	Searching for and expanding ideas	Classifying and analyzing information	Generating and evaluating alternatives	Preliminary design production and construction	Post completion evaluation
Collective Decision Methods						
Brainstorming	●	●				
Buzz sessions	●					
Group discussions	●	●				
Role play	●	●				
Synectics	●	●				
Comparison Methods						
Paired comparisons			●	●		●
Ranking and weighting			●	●		●
Preference matrix				●		●
Evaluation matrix				●		●
Trade-off games	●	●	●			
Rating Methods						
Rating scale	●			●		●
Guttman scale	●					●
User rating test	●					●
Building performance test				●		●
Semantic rating test	●			●		●
Spatial performance test	●			●		●
Visual Preference Methods						
Visual preference				●		●
Spatial preference				●		●
Attribute discrimination						
Checklists						
Code and zoning checklist			●	●		●
Activities checklist	●		●	●		●
Descriptive and Evaluative Methods						
Behavioral mapping				●		●
Social mapping(sociogram)				●		●
Activity log	●					●
Design Methods						
Activity analysis			●	●		
Pattern language				●		
Performance method				●		
Morphological method				●		
Systems method			●	●		

Figure 13. The Programmer's Kit of Parts (13:92)

The book partitions the techniques into (1) information retrieval methods, and (2) methods of transforming design information. In addition, the techniques are classified by general "method" or procedures. (13:92)

Programmers and design professionals have developed techniques to fit the unique information needs of facility construction. However, many techniques are borrowed from other areas of interest, such as: (1) management science, (2) statistics, (3) market and opinion research, (4) behavioral science, (5) social science, (6) computer science, (7) communications, and (8) planning (17:11). The following examines different categories of techniques.

Research Techniques. Research techniques are used primarily to collect programming information. They are the most traditional and familiar means of gathering data. Primary sources of programming information are the personal knowledge, experiences, and perceptions of the client, owner and user. Research techniques collect this information in the form of opinions, attitudes, descriptive data and evaluative data. The techniques included are:

1. Background Data Research
2. Surveys
3. Interviews
4. Questionnaires
5. Data Logs
6. Standardized Data Forms.

The above techniques are basic tools in any data collection effort. At least one is essential to any programming effort, though, a combination of techniques is more effective. (17:53-55)

<u>Observation Techniques</u>. Another group of data collection tools are observation techniques. They are direct and reliable means of gathering behavioral information. Simply, using these techniques, programmers observe people in their physical and social environments. The type of information collected include how individuals or groups behave in or react to their surroundings. Observation techniques are valuable tools since they can discover new information and verify information collected by other means. However, programmers should use them to supplement other programming techniques since they can not adequately identify project needs by themselves. Their usefulness is limited to identifying behaviors in existing conditions and can not predict behaviors to new environments. (17:70-72)

A list of observation techniques include:

1. Direct Observation
2. Tracking
3. Participant Observation
4. Behavior Mapping
5. Behavior Specimen Record
6. Instrumented Observation

__Comparison Techniques__. Comparison techniques are used primarily for the collection and analysis of information. The various methods compare "statements or concepts to determine orders of preference and desirability (13:60)." They are also referred to "attitude measurement" techniques because they quantify individual or group values, feelings, perceptions, priorities, preferences, and goals. One important function of comparison techniques is to clarify user attitudes compared to that of the programmer and designer. This helps prevent the programmer or designer from imposing his own values or preferences during the project development. (13:60; 17:79-80)

A list of comparison techniques include:

1. Paired Comparisons.
2. Ranking Chart.
3. Preference Matrix.
4. Evaluation Matrix.
5. Trade-Off Games.
6. Adjective Checklist.

Comparison techniques include some sophisticated methods. Many programming efforts may not require the use of these techniques. Often, research techniques, such as interview or surveys, are adequate in identifying individual or group attitudes. Also, experience in psychology, sociology, and statistics are recommended when using these techniques. (17:79-80)

Statistical Analysis. Statistical analysis refers to the mathematical processes used to quantify information or variables. The use of the numerical data allows measurement, differentiation, and correlation of variables. There are basically two classifications of statistics: (1) descriptive, and (2) inferential. The first, descriptive statistics, are relatively simple procedures that produce averages, percentages, distributions, and variances. The second, inferential statistics, usually involve more complicated techniques such as factor analysis, regression analysis and analysis of variance. The later procedures are useful in predicting future outcomes or drawing conclusions based on sample data. Programmers can use statistics to:

1. Simplify the description and calculation of factors.
2. Reduce mixed variables to a common quantifying basis for comparison and correlation.
3. Test the validity and reliability of data and conclusions.
4. Predict the varying impacts of problem components on each other and on the whole problem.
5. Optimize elements and combination of elements.
6. Improve precision of calculations. (17:88)

Statistical analysis techniques are useful in wide variety of areas, such as measuring attitudes, evaluating alternatives, and projecting future needs, among others. (17:88-89)

Functional and Activity Analysis. Techniques that analyze the client's functions are important tools in programming. A client's organization is based on an "operational system of activities and relationships that is organized for the accomplishment of specified objectives (17:94)." A facility, on the other hand, exists to "facilitate" this operational system. In other words, understanding the a project's functional needs is essential to creating a physical system (the facility) that enhances or improves the operational system (17:94). Functional or activity analysis techniques:

> 1. Identify functional or activity components.
> 2. Assess relevant dimensions or attributes of individual components.
> 3. Rate or rank components according to relative significance and organizational status.
> 4. Identify relationships among components.
> 5. Group components in according to interdependencies.
> 6. Establish performance goals, requirements or criteria.
> 7. Resolve conflicts among components.
> 8. Organize or reorganize components into an efficient, effective system. (17:94-95)

Space Analysis. Space is described as "the single most important element of a facility (17:99)." In fact, all other programming elements depend on the physical characteristics of space. Space analysis techniques are then a crucial component of most programming efforts. (17:99)

> The purpose of space analysis is to determine the physical characteristics - the quantity and conditions - that can accommodate the a client's objectives, philosophy, organization, and activities. (17:100)

Space analyses can include:

1. Identification of appropriate units of space.
2. Space unit requirements.
3. Space inventory.

4. Equipment/furnishings inventory.

5. Space plan or layout.

6. Space program summary.

7. Space program.

8. Space budget or cost estimate. (17:100)

__Cost Analysis__. Cost analysis techniques are often part of facility programming to estimate construction costs, and facility operation and maintenance costs prior to design. The main benefit of including cost analysis techniques in programming is to determine project feasibility within funding limitations. For example, once design is complete the proposed project's costs may be prohibited. Facility redesign or project loss are possible outcomes. By including cost analyses in the programming phase, the client can avoid these undesirable consequences. (17:112-113)

Types of cost analyses include:

1. Project cost estimating.

2. Construction cost estimating.

3. Life-cycle cost analysis.

4. Cost-benefit analysis.

__Scheduling Techniques__. Scheduling is an important aspect of any facility project. Scheduling estimates the amount of time and sequence of events or activities. In order to complete a project in an efficient, cost-effective manner, projected schedules are composed. For a programmer, a schedule may forecast the time and arrangement of programming activities. Also, a programmer may construct schedules for design, construction and occupancy during the programming effort. In addition, schedules are dynamic tools subject to revision and change as projects develop. They can help assess the project's progress and identify critical areas where time is a factor. (17:115-117)

Types of schedules include:

1. Programming schedules.
2. Project schedules.
3. Design and/or construction schedules.
4. Occupancy schedules.
5. Projected use schedules.
6. Master plan or development schedules.
7. Site development schedules.

Major components of any successful scheduling effort include "clear identification of the necessary tasks, accurate estimates of their time requirements and well-planned coordination of work performance (17:117)."

Types of scheduling techniques include:

1. Bar charts/milestone charts.
2. Activity time Chart
3. Critical path method (CPM).
4. Program evaluation and review technique (PERT).
5. Precedence diagraming method (PDM).

Relationship Matrices. One of the most widely used tools for organizing programming data is a relationship matrix. Relationship matrices identify, define and measure facility user, space or activity interactions. They can convey existing or desired relationships. A matrix is a visual tool that can quickly show "individual interactions in relation to the total set of interactions (17:121)." The relationships matrices can identify and organize are (1) functional, (2) organizational, (3) space, and (4) activity. (17:121)

Matrices are used to:

> 1. Collect and record data directly about relationships, as in a questionnaire or interview.
> 2. Enumerate possible combinations of factors and isolate significant combinations.
> 3. Analyze previously determined relationship data.
> 4. Summarize optimum relationship data.
> 5. Communicate conclusive data.
> 6. Describe existing conditions or predict desirable relationships.
> 7. Initiate more sophisticated analysis of relationships. (17:121-122)

<u>Correlation Diagrams</u>. Correlation diagrams are another way to organize programming data. Like matrices, they deal primarily with patterns of relationships. Their main purpose is to graphically "depict functional and space relationships (17:123)." In fact, correlation diagrams and relationship matrices are said to be complementary. Correlation diagrams are often based on data from a matrix. These diagrams visually interpret relationships for analysis, evaluation and communication. Figure 14 lists thirteen correlation diagrams by their type of relationship and form of representation.

	Relationship				Representation		
	Functional	Physical	Social	Factor	Abstract	Pictorial	Scaled
Social Map		●	●		●		●
Sociogram			●		●		
Behavior Map	●	●	●		●		●
Bubble Diagram	●	●		●	●	●	
Link-Node Diagram	●		●	●	●	●	
Block Diagram	●	●			●	●	
Interaction Net	●		●	●	●		●
Dual Graph		●		●	●		●
Adjacency Diagram		●				●	●
Functional Relationship Diagram	●				●	●	
Layout Diagram		●					●
Flow Diagram	●			●		●	
Organizational Chart	●		●			●	

Figure 14. Type of Relationship and Representation of Different Correlation Diagrams (17:127)

__Collective Decision Techniques__. Collective decision techniques are communication devices for group decision making. Often in a project, many interested parties are involved. These individuals or groups may all have valuable input, but may also have conflicting interests. Collective decision techniques are methods that can aid in generating new ideas and alternatives. In addition, they can resolve conflicts and facilitate client consensus. These techniques are also called "participation interaction" methods, because they are ways of involving the owner, user and client in programming. (13:14; 17:136)

The list of collective decision techniques are:

1. Brainstorming.
2. Synetics.
3. Buzz/rap sessions.
4. Role playing.
5. Gaming.
6. Group planning.

<u>Documentation/Presentation Techniques</u>. Once the programming information in collected, analyzed, organized and evaluated, it must be communicated. Documentation/ presentation techniques are ways of conveying programming conclusions to the client, designer, or any other concerned party. The three basic methods include: (1) printed narratives, (2) audio-visual presentations, and (3) oral presentations. The narrative is useful in two respects. First, it can be used as a reference for the designer during design. Second, it can solicit client or owner approval prior to initiating design. The narrative, also, allows the careful selection of words and phrases geared towards the intended audience. On the other hand, audio-visual presentations are "more stimulating and memorable" but are "costly and time consuming to prepare (17:140)." Oral

presentations combine some of the advantages of both the narrative and the audio-visual presentations. In addition, they often also allow audience participation in the form of questions. Also, the programmer can more quickly prepare an oral presentation than either a narrative or audio-visual presentation. Graphics are another primary technique in documentation and presentation of programming data. They are often found in narratives, audio-visual presentations, and oral presentations. Graphics allow the audience the rapidly comprehend information that words may not convey easily. (17:140-141)

Rating Techniques. Rating techniques are evaluation tools "for judging the value, reliability or appropriateness of data, conclusions and options (17:149)." The evaluators include the client, user, designer and programmer. However, rating techniques are often geared towards using the client's or user's experience to assess some aspect of the programming information. Major advantages of rating techniques are quick and reliable gathering of input. The main objectives of these tools are (1) problem identification and exploration, and (2) generation and evaluation of alternatives. (17:149; 13:70)

A list of rating techniques include:

1. Rating scales.
2. Guttman scales.
3. User rating tests.

4. Building performance tests.

5. Semantic rating tests.

6. Spatial performance tests.

Computer-Aided Facility Management

Many of the programming techniques, mentioned previously, can benefit from using computers. Computers can handle large amounts of data, and they are quick, efficient, reliable and precise. One particular area of facility programming, space analysis, has seen a proliferation of software programs called Computer-Aided Facility Management (CAFM). There are over 60 CAFM programs on the market today. CAFM, usually, focuses in one of two areas: (1) facility maintenance, or (2) space considerations. The concentration on space analysis is significant since "determining the amounts and kinds of spaces required for an architectural project is a fundamental function of programming (17:163)." Currently, many programmers use CAFM programs in their work. (28:68)

The researcher had the opportunity to review one CAFM program called, FM:Space-Management, developed by FM:Systems. The software provided some potentially valuable features for space analysis, such as (1) forecasting future space needs, (2) generating stacking and blocking solutions, and (3) tracking space inventory. The program, also, interfaced with two different CADD (Computer-Aided Drafting and Design) systems. (29)

Programming Strengths and Weaknesses

In closing the literature review, the researcher wanted to focus on the strengths and weaknesses of facility programming, or what does and does not produce a "good" program. White reported on the following areas in his research: (1) key programming skills, (2) programming strengths, and (3) programming problems. First, the results of the research revealed that a programmer should be or have skills in following areas:

1. Communication
2. Information Processing
3. Design/Building Delivery
4. Human Relations
5. Synthesizing and Concluding
6. Inventive and Creative
7. Attention to Detail
8. Graphics (30:24)

Second, when the research participants were asked "what they were most proud of about the way they programmed their jobs," the following were listed as programming strengths.

1. Thorough, rigorous, analytic process
2. Strong client/user participation
3. Programming tailored to each project
4. Strong interaction with design
5. Successful projects/happy clients
6. Good communication
7. Program not an end but a means (30:24)

Third, the following were named as areas of difficulty in programming.

1. Finding the true needs of the client
2. Getting clients to make decisions
3. Clients don't appreciate programming
4. Sloppy prior programming
5. Program-design connection
6. Changes of mind
7. Programs done by consultants
8. Staffing the programming phase (30:25)

In the May 1988 issue of Architecture, an article reported the results of interviews with some prominent programmers. One question asked was: "What makes a good program?" Some of the answers included were:

1. Clarity of communication.

2. Description of each space's function.

3. Justification of users' behaviors, needs, and satisfactions.

4. Identification of goals and functions.

5. Regard to the site, surroundings and context.

6. Including of design ideas. (14:206)

Chapter Summary

The literature review examined many aspects of facility programming including: (1) the purpose of programming, (2) benefits of programming, (3) the programming - design interface, (4) programming techniques, and (5) programming strengths and weaknesses. The information collected was important to the next phase of research, the Delphi method, by providing the framework, and content validity for the survey instruments. The next chapter, methodology, was built on the literature. It includes the rationale for the research design, and how the research was carried out.

III. Methodology

Chapter Overview

This chapter presents the research steps that address the problem statement, and research objectives and questions. The researcher gives a description of the research design, addresses the importance of the literature review, and explains the steps of the Delphi method as applied to this research. The chapter, also, contains details of the participant selection, questionnaire design, and administration processes used by the researcher.

General Description

The research was designed to the solve the problem of developing a better facility programming process for the Air Force. The research followed the widely used rational decision-making process that includes five steps: (1) diagnose the problem, (2) find alternative solutions, (3) analyze and compare alternatives, (4) select an alternative, and (5) implement the solution. The research design included two primary data collection techniques: (1) the literature review, and (2) the Delphi method.

The Literature Review

An important first step in this research was the literature review. Presented in chapter two of this study, a comprehensive literature and data search was conducted of professional journals, periodicals, and books that document and explain programming methods and theory. The literature review helped determine the content validity of the research by defining the basic assumptions and bounds of the research. Key issues were identified and studied in the next phase of research, an interactive survey process using the using the Delphi method (31:15).

The Delphi Method

The RAND corporation developed the Delphi method in the late 1940's to solicit and organize consensus, expert opinion. The key objective is the consensus of participants by a "controlled and rational exchange of iterated opinion (31:6-7)." The conventional Delphi technique exhibits the following characteristics:

1. The participants are usually experts in the field of study.
2. The data collection format is typically a structured formal questionnaire.
3. The questionnaire contains items, quantitative or qualitative, about the study's objectives.
4. The questionnaire items are generated by the researcher, participants, or both.
5. A set of instructions, guidelines, and ground rules accompany the questionnaire.
6. The questionnaire is administered to the participants for two or more iterations.
7. The participants answer scaled questions and/or requests for written responses.
8. Statistical feedback and/or selected written responses accompany each iteration of the questionnaire.
9. Individual responses to all iterations are kept anonymous.
10. The researcher may ask outliers (i.e. upper and lower quartile responses) to justify their responses in writing.

11. The iterations with feedback continue until the participants reach a consensus, as determined by the researcher.

12. Participants do not discuss issues face-to-face (31:7).

The Delphi technique takes advantage of the (1) knowledge and judgment of experts, (2) the group decision making process, and (3) transfer of information during feedback. The Delphi, also, reduces the disadvantages of group interaction with three key features: (1) anonymity, (2) controlled feedback, and (3) statistical group response (32:3). Anonymity helps eliminate problems with face-to-face group discussions, such as:

> the presence of a dominant, persuasive personality, the tendency to want to meet the approval of the group and the unwillingness to change an opinion which had been publicly expressed. (33:2)

Controlled feedback cuts down on "noise," another problem with group interaction. Noise is defined as "irrelevant or redundant material that obscure the directly relevant material offered by participants (32:3)." The last attribute, statistical group response, further lessens group pressure to conform since there is no "particular attempt to arrive at unanimity among the respondents (32:3)."

Application of the Delphi Method

The research applied the Delphi method to pool expert opinion on the facility delivery process with specific attention to the programming phase. The Delphi technique includes five steps: (1) establishing the objectives, (2) selecting the participants, (3) designing the questionnaire, (4) administering the questionnaire, and (5) interpreting the results (31:3). The following describes each step in general terms.

Establishing the Objectives. The objectives of the research are:

1. Identify the weaknesses and strengths of the programming processes used by the Air Force.

2. Identify the weaknesses and strengths of the programming processes used by commercial Architect-Engineering firms.

3. Combine the successful elements into a new programming model.

4. Recommend ways to test and validate the new programming model.

Selecting the Participants. The Delphi method relies on the knowledge and judgment of experts. However:

> The selection of experts is an intricate problem even when the category of expertise needed is well-defined. A man's experience might be judged by his status among his peers, by his years of professional experience, [or] by his own self-appraisal of relative competence in different areas of inquiry. (33:4)

The participants in the research are "experts" in facility programming. The following describes the universes, populations, and sample of participants.

The Universes. The first universe for this research consists of all Air Force Civil Engineering personnel, military or civilian, who are facility programmers. These personnel may be located at the Air Staff, the MAJCOMs, the AFRCEs, or Base Civil Engineering organizations.

The second universe consists of all architects who are facility programmers. They may be located anywhere in the United States.

<u>**The Populations**</u>. The populations of interest are a group of "experts" in facility programming either working for the Air Force or a commercial Architect-Engineering firm. An "expert" in facility programming will be defined by expertise, years of professional service, and status among his peers.

<u>**The Samples**</u>. The first sample consists of architects working as facility programmers. Architects were chosen for the first sample because:

> The professional architect, by training and experience, is not only able to assimilate and translate the wants and requirements of a client, but to combine that information with the architectural and other requirements for design, of which the client is often unaware. (17:14-15)

Chief Engineers at Base Civil Engineering organizations constituted the second sample of participants. Chief Engineers were selected because they generally supervise the entire facility delivery process, which includes programming.

The research plan was to identify 15 to 20 participants for each of the samples. A more detailed account of the participant selection process is given later in the chapter.

__Designing the Questionnaire__. The researcher drafted two questionnaires for each Delphi round, one for each sample of participants. One reason includes the operational differences between the Air Force and private industry. The literature review has discovered significant variations in the programming processes and terminologies used by each research population.

The questionnaires were the primary data gathering tools in this research. The researcher tailored the instruments to provide two types of information: (1) answers to the research questions, and (2) classifications of the respondents.

The form for the questionnaire encouraged both open (free choice of words) responses and required closed (specified alternatives) responses. The open-responses were included to gather more detailed information on how the respondents felt about questions. The closed-response questions were used because the respondents are experts with a clear understanding of the topic (34:217). For classifications of the respondents, the researcher used use multiple choice questions. The researcher used four major decision areas in developing a survey instrument: (1) question content, (2) question wording, (3) response form, and (4) question sequence (34:207). A more detailed description of each questionnaire design is provided later in the chapter.

Administering the Questionnaire. The researcher administered the questionnaires through the Air Force distribution system and the U.S. Postal Service because of the expected wide dispersal of the respondents throughout the United States. Studies of the Delphi method indicate consensus on questionnaire items occur by the second or third iteration, if at all. Consequently, the researcher planned for three rounds to reach final consensus on the research questions. However, only two rounds were administered due to time constraints. Again, a more detailed account of the questionnaires administration is given later in the chapter.

Interpreting the Results. The final stage is the write up and dispersal of the results. Because the Delphi method is an iterative process, an analysis of each round of questionnaires is required. The first round included evaluating data from the questions concerning the research objectives and the respondent characteristics. Round two involved only the analysis of items answering the research questions. Interpretation of the results included both statistical tests and personal judgments by the researcher.

Criteria for Consensus. The main objective of the Delphi technique is consensus of participants on an issue. The researcher provided questions on a five-point Likert scale and in multiple-choice format. Criteria for consensus was set for each type of question and for each round. The criteria is discussed further in Chapters IV and V.

Statistical Tests. After the first questionnaire and each successive questionnaire, the Delphi method requires statistical feedback of results to the respondents. The feedback usually "involves a measure of central tendency, some measure of dispersion, or perhaps the entire frequency distribution of responses for each item (32:7)." The researcher used descriptive statistics to measure the above items including: (1) frequency distributions, (2) percentages, (3) means, (4) medians, and (5) standard deviations.

The research design contained two distinct populations, namely Air Force personnel and commercial architects who program facilities. The researcher employed non-parametric statistics to detect any significant differences between the two groups concerning facility programming. Specifically, the Wilcoxon-Rank Sum Test was used, because (1) the researcher could not assume normal samples, and (2) the test is "at least 86 percent as efficient as the t-test (35:613)."

Interpretations by the Researcher. The researcher's role in the Delphi technique is critical because he selects the types and amounts of feedback in the subsequent rounds of questionnaires. In addition to statistical data, the researcher must interpret written responses by the participants. The researcher must temper his own biases when using his judgment.

Participant Selection

The research involved two groups of participants: (1) architects with experience in facility programming, and (2) military employees with experience in facility design and construction. For the purpose of the research, the sample populations will be referred to as Group A and Group B, respectively. The participants were selected in the following ways.

Group A. The researcher selected the Group A participants primarily through the literature review and personal references. The literature review included books, periodical articles and other research on facility programming. Individuals who either wrote or were interviewed on the subject matter were invited to participate in the research. During the participant search, several people were named as "experts" in the field of study. These individuals were also asked to participate. The researcher made approximately 40 telephone calls over a two month period (February to April 1990) to solicit participation in the study. The researcher contacted 25 potential "experts" directly and all agreed to participate in the research. During the telephone conversations, the researcher explained the purpose of the research, the proposed research method, and the estimated time required from each participant. The researcher took care to select individuals from throughout the continental United States to account for any geographical differences. One participant works and resides in Canada. The Group A participants are architects with experience in facility programming either working in private firms or educators at major universities. Appendix A is a partial list of the Group A participants. Their individual names and professional associations are printed with their written permission. The list contains the names of many prominent authors and researchers in the programming field.

Group B. The researcher solicited 40 Chief Engineers working in Air Force Civil Engineering squadrons to participate in the research. Chief Engineers were chosen as Group B participants because of the their expertise in the Air Force facility design and construction process. The Chief Engineer is in charge of the Engineering Branch that usually includes four functional sections: (1) Contract Programming and Environmental Planning, (2) Design, (3) Contract or Construction Management, and (4) Real Property. Air Force facility programming is typically handled within the Engineering Branch. Though the Chief Engineer does not directly program facility projects, as the branch chief, he is familiar with the methods and process used by the Air Force. The Chief Engineer, also, has the advantage of "seeing" how programming interacts with design and effects construction.

The 40 participants were randomly selected from a list of 77 Civil Engineering squadrons located at Air Force bases within the continental United States (CONUS). The final list was slightly adjusted to include a mixture of bases from the major commands and to account for any geographical differences (Table 1). The participants were not contacted prior to mailing the first round questionnaire. The initial survey package did include a letter requesting participation in the research, specifying the purpose of the research, and explaining the research method. The researcher's goal was a 50 percent response rate for 20 "experts."

TABLE 1

GROUP B PARTICIPANTS BY MAJOR COMMAND

	NUMBER OF PARTICIPANTS	
MAJOR COMMAND	ROUND ONE	ROUND TWO
SAC	12	9
MAC	7	5
TAC	7	4
ATC	2	0
AFLC	1	1
AFSC	1	1
AFSPACECOM	1	0
TOTAL	31	20

Round One Questionnaire Design

The Delphi questionnaires for the two groups were formulated to support the research questions and the findings of the initial literature review. Both questionnaires for Group A and Group B were similar. The researcher broke the survey into five parts as follows.

Demographic Questions. Questions 1 - 5 requested information on the respondents' educational backgrounds and their experience in programming, design and construction management. Questions 1 - 4 were exactly the same in both surveys. Question 5 for Group A dealt with the type of services the respondents or their firms provide. Question 5 for Group B dealt with the respondents' experience in Air Force Civil Engineering. The demographic questions were included to support the participants' "expertise" in programming and the facility acquisition process.

Rated Scale Questions. Questions 6 - 40 focused on (1) facility programming methods, (2) programming's part in the facility delivery process, (3) programming's interaction with facility design, and (4) the roles of key players in programming. The researcher provided a five-point rating scale for the participants' answers to each question. The questions were written in the form of statements with the responses ranging from "strongly agree" to "strongly disagree." 27 of the 35 questions in the Group A and Group B questionnaires were identical. The remaining 13 questions were similar in content and wording except for several terms. In the Group A survey the terms "client/owner", "client", or "client/user" were used in these questions. In the Group B survey, the Group A terms were substituted with "user/using agency". The distinction in terms were to account for operational differences between the two groups of respondents.

Multiple Choice Questions. The questionnaire design included two types of multiple choice questions: (1) single response and (2) multiple response. Questions 41 - 50 were written in the single response format. Single response is defined as questions with multiple, mutually exclusive answers. The researcher added these questions to support, clarify and validate various key questions in the Rated Scale portion of the questionnaire. The researcher requested the participants give only one response to each question. Most of the questions in the Group A and Group B questionnaires were identical. However, in the Group A questionnaire the term "client" was used in two questions. In the Group B questionnaire, the term "client" was substituted with "user/using agency" in the same two questions. Again, the distinction in terms were to account for operational differences between the two groups.

Questions 51 - 54 requested information on programming content and specific programming techniques. This section included questions with multiple responses. The researcher requested the participants "mark" all applicable answers to each question. All four questions were identical in the Group A and Group B surveys.

Open-Ended Question. Question 55 in both questionnaires was an open-ended question that requested a written response from the participants. However, the question was different for each group. Group A was asked: "What two or three questions would you like to ask your peers about facility programming?" Group B was asked: "Do you believe Air Force programming methods adequately define project requirements prior to initiating design?"

Round One Questionnaire Administration

The round one questionnaires were distributed through the U.S. Postal Service and the Air Force distribution system. The survey packages included: (1) a personal letter, (2) general instructions, (3) the survey instrument, and (4) a pre-addressed return envelop. The envelops for Group A, also, included postage to further encourage a high response rate. Return postage for Group B was provided by the Air Force distribution system. The correspondence (1) was on Air Force Institute of Technology letter head and personally signed by the researcher. The letter requested participation in the research, specified the purpose of the research and explained the research method. For Group A, the letter was personally addressed to the participant. Group B's letters were addressed to the Chief Engineer at the individual Air Force base. On 9 May 1990, 25 questionnaires were mailed to the Group A participants. The Group B survey packages were sent a week later on 17 May 1990. In addition, follow-up telephone calls were made to respondents, if necessary, to increase the response rate. Appendix B contains copies of the correspondence, instructions and questionnaires.

Round Two Questionnaire Design

The round two Delphi questionnaires reexamined questions asked in the first round. However, the format for the round two questionnaire was different from the first round. Round one questions were grouped according to the kind of question (i.e. demographic). In round two, the research instrument was broken into five sections. Each section contained 3 to 6 related questions with the appropriate data from the first questionnaire. In other words, the questions were sequenced according to general topic area, not by response form.

Questionnaire Content. First, no new questions were added to the research effort. The round two questionnaires repeated questions from the first survey. However, approximately half the questions were eliminated from both round one questionnaires. All five demographic questions, three multiple choice (single response) questions, one multiple choice (multiple response) question and the open-ended question were eliminated from round two because the researcher determined the responses were based on factual information. In other words, the researcher should obtain the same responses regardless of any input. However, rated scale and multiple choice (single response) questions were subject to round one consensus criteria. As a result, if the respondents reached consensus on the question, it was not included in the new survey. In the end, the questionnaires included all the remaining questions.

Though the format for the Group A and Group B questionnaires for round two were similar, their content was not the same. Since many of the questions were subject to consensus criteria, the two questionnaires do not contain all the same questions. In other words, the Group A and Group B respondents did not necessarily reach consensus on the same questions. As a result, number and sequence of questions is different for both questionnaires.

Statistical Feedback. The second round questions included statistical feedback from the first questionnaires. All the questions were evaluated based on frequency of responses. In addition, responses to the rated scale questions were given a numerical value as follows:

RESPONSE	VALUE
A. Strongly Agree	5
B. Agree	4
C. Undecided	3
D. Disagree	2
E. Strongly Disagree	1

The numerical values allowed the researcher to calculate descriptive statistics, such as the mean and median, for each question. The responses to multiple choice questions did not receive numerical values. As a result, each rated scale question included the following data: (1) the frequency of each response, (2) the percentage of each response, (3) the number of responses, (4) the mean (or average) response, and (5) the median (or middle) response. While, each multiple choice question included the following information: (1) frequency of each response, and (2) the percentage of each response.

Respondent Comments. The round one questionnaire instructions encouraged written comments to all the questions. The comments were invaluable in determining question content and construct validity, and reliability. As a result, many of the questions were changed or clarified in the round two questionnaires.

In addition to statistical feedback, the round two questionnaires included the first round written comments. At the end of the five sections, comments pertaining to each section question were annotated. The comments were included because they justified or clarified a respondent's view on an issue.

Changes and Clarifications. In response to respondent comments, the second round questionnaire included additions, omissions and definitions of words or phrases contained in particular questions. The changes and clarifications were annotated on the questionnaire, as follows:

1. Additions. Words or phrases added to a question were italicized.

2. Omissions. Words or phrases omitted from the question, but were included in the first questionnaire were bracketed.

3. Definitions. Words or phrases that were defined in each section were bold-faced.

The changes were annotated for several reasons: (1) to give context to the responses given in the original version of the question; (2) to clarify the intent of the question; (3) to clarify the meaning of a word or phrase.

Round Two Questionnaire Administration

The round two questionnaire administration was similar to the round one procedures. The survey packages were distributed through the U.S. Postal Service and Air Force Distribution system. The packages included: (1) a personal letter, (2) general instructions, (3) instructions on how to read the questionnaire, (4) the survey instrument, and (5) a pre-addressed return envelop. Again, return postage was included for Group A, because they were outside the Air Force distribution system. The correspondence (1) was on Air Force Institute of Technology letter head personally signed by the researcher. The letter requested participation in the second round questionnaire and thanked the respondents for their participation in the first round questionnaire. The letter, also, included information on the round one response rate, summarized the respondent group's characteristics based on their responses to the demographic questions, and explained the round one criteria for consensus. In addition, the researcher included two pages of instructions called "How to Read the Questionnaire." The detailed instructions explained the questionnaire's format and content which included (1) the questions (2) respondents' comments, (3) statistical data, and (4) changes and clarifications to questions. The Group A and Group B survey packages were mailed on 14 June and 23 June 1990, respectively. Follow-up phone calls were made to participants when necessary. Appendix C contains copies of the correspondence, instructions and questionnaires.

Written Responses

In closing the methodology chapter, a discussion of the unstructured written responses is necessary. In both Delphi rounds the respondents were encouraged to justify or explain their answers to the structured questions. The written comments were valuable in interpreting the underlying attitudes about programming issues that the statistical data could not reveal. The comments from the four survey instruments are reproduced in Appendices D, E, F, and G.

Chapter Summary

The research design used two primary data collection techniques: (1) the literature review, and (2) the Delphi method. The literature review was important because in establishing content validity. The Delphi method solicited expert opinion with the goal of group consensus.

The research applied the Delphi method to answer the research questions by pooling expert opinion in the field of facility programming. The Delphi technique included five steps: (1) establishing the objectives, (2) selecting the participants, (3) designing the questionnaire, (4) administering the questionnaire, and (5) interpreting the results.

The research design identified two samples of participants, Air Force personnel and architects working as facility programmers. The questionnaires were the primary data gathering tool in the research. The questionnaires solicited "expert" opinion through two or more iterative questionnaires to reach a consensus on an issue. Interpretation of the results included both statistical tests and personal judgments by the researcher.

The next two chapters report the results from the Delphi questionnaires. They include a narrative accompanied by the statistical data in tabular form. The information includes whether the groups reached consensus on a question based on their responses.

IV. Round One Delphi Questionnaire Results

Chapter Overview

This chapter reports the results of the first round questionnaires for the two research groups. The Group A and Group B survey instruments each contained 55 questions. The resulting data is broken into 5 broad categories for review: (1) Respondent Experience, (2) Programming Content, (3) Programming Participants, (4) Programming and Design Interaction, and (5) Programming Techniques. The chapter narrative is accompanied by the statistical data presented in tabular form comparing the two groups of respondents.

General Results

As previously mentioned in Chapter III, the researcher's objective was a total of 20 participants from each respondent group. This goal was achieved in the first round.

Group A. 22 of the 25 questionnaires were completed and returned over a six week period. The response rate was 88 percent. Consensus was reached on 19 of the 41 applicable questions in the first round.

Group B. 31 of the 40 questionnaires were completed and returned over a six week period. The response rate was 77.5 percent. Consensus was reached on 18 of the 41 applicable questions.

Criteria for Consensus

The main objective of the research method, the Delphi Technique, is the consensus of respondents on an issue or question. For the purposes of the Round One questionnaires, the criteria for consensus for multiple choice and rated scale questions was:

Multiple Choice. A 70 percent agreement among respondents on a single answer, multiple choice question constituted consensus.

Rated Scale. An 80 percent agreement among respondents on rated scale questions constituted consensus based on two groups of responses: "strongly agree/agree" and "strongly disagree/disagree."

The researcher used conservative numbers to determine consensus on the first round questionnaires. For the purposes of the final analysis, consensus criteria is 10 percent lower (60 and 70 percent) for the multiple choice and rated scale questions, respectively.

Respondent Experience

Questions 1 - 5 establish the Group A and Group B participants' experience and expertise in the facility delivery process. Question 1 requested information on the respondents' educational backgrounds (Table 2). The Group A participants all had educational backgrounds in architecture. In contrast, the overwhelming majority (96.8%) of the Group B participants had educational

TABLE 2

EDUCATIONAL BACKGROUNDS OF RESPONDENTS

EDUCATION	GROUP A	GROUP B
1. Architecture	19	1
2. Architecture/Planning	2	0
3. Architecture/Psychology	2	0
4. Architecture/ Civil Engineering	0	1
5. Civil Engineering	0	18
6. Civil Engineering/ Sanitary Engineering	0	1
7. Mechanical Engineering	0	5
8. Mechanical Engineering/ Executive Development	0	1
9. Electrical Engineering	0	3
10. Agricultural Engineering	0	1
SAMPLE SIZE	23	31

backgrounds in engineering. Only 2 individuals from Group B had formal educations in architecture. The Group B participants' backgrounds were divided among several engineering disciplines. Of the Group B respondents, 20 (64.5%) had educations in civil engineering.

Questions 3 - 4 dealt with the years of experience the participants had in (1) programming, (2) design, and (3) construction management, respectively (Table 3). All Group A respondents had experience in programming with 82.6 percent having 10 or more years of experience. None of the individuals in Group A had less than 5 years of programming experience. 26 participants in Group B had experience in programming. Of the Group B respondents, a majority (54.8%) had 8 or more years of experience in programming.

TABLE 3

PROFESSIONAL EXPERIENCE OF RESPONDENTS

CATEGORY/YEARS	GROUP A	GROUP B
PROGRAMMING		
None	0	5
Less than 5 Years	0	4
5 to 7 Years	2	5
8 to 10 Years	2	4
11 to 13 Years	5	4
14 or More Years	14	9
DESIGN		
None	2	1
Less than 5 Years	2	4
5 to 7 Years	1	5
8 to 10 Years	1	3
11 to 13 Years	4	1
14 or More Years	13	17
CONSTRUCTION MANAGEMENT OR INSPECTION		
None	9	3
Less than 5 Years	4	8
5 to 7 Years	1	5
8 to 10 Years	2	2
11 to 13 Years	1	1
14 or More Years	6	12
AIR FORCE CIVIL ENGINEERING		
None	NA	0
Less than 5 Years		1
5 to 7 Years		2
8 to 10 Years		4
11 to 13 Years		5
14 or More Years		14
SAMPLE SIZE	23	31

NA - The question was not applicable.

In Group A, 21 of 23 respondents said they had some experience in design. In addition, 73.9 percent of the Group A participants had 10 or more years of experience in design. In comparison, 30 of 31 Group B participants responded they had some experience in design. In Group B, 67.7 percent of the respondents had 8 or more years of design experience.

In Group A, only 14 of the 23 participants (60.8%) had experience in construction management or inspection. In contrast, 90.3 percent of the Group B respondents had construction management or inspection experience. However, only 48.4 percent of Group B participants showed 8 or more years of experience in construction management.

Clearly, both Group A and Group B have strong backgrounds in design. The differences between the groups occur in the programming and construction management areas. Using years of experience as an indicator, Group A demonstrates concentrated "expertise" in programming. The researcher expected such a result based on the participant selection procedure. Group A, however, only shows evidence of moderate experience in construction management. In comparison, Group B has a weaker base of experience in programming and a stronger base of experience in construction management. The researcher, though, classifies Group B's "expertise" in programming and construction management both as moderate.

Question 5 in the Group A and Group B questionnaires were different. For Group A, information on the type of services the respondents or their firms provide was requested (Table 4). The data revealed 11 different services provided by the group. Almost all the respondents (95.6%) provided programming services and a majority of respondents (69.6%) provided architectural design services. The next largest service, engineering design, was only listed by 7 respondents (30.4%).

Question 5 for Group B asked for the years of experience working in Air Force Civil Engineering organizations (Table 3). As expected, all Group B respondents have experience in Air Force Civil Engineering. In addition, 74.2 percent of the Group B participants have 8 or more years of experience.

TABLE 4

SERVICES PROVIDED BY RESPONDENTS
OR RESPONDENTS' FIRMS (GROUP A ONLY)

SERVICE	FREQUENCY	PERCENTAGE
Programming	22	95.6
Architectural Design	16	69.5
Engineering Design	7	30.4
Post-Occupancy Evaluation	4	17.4
Interior Design	3	13.0
Construction Management	2	8.7
Master Planning	2	8.7
Urban Planning	1	4.3
Standards Development	1	4.3
Software Development	1	4.3
Research	1	4.3

Programming Content

The researcher included 8 questions dealing directly with the type of information provided with facility programming. In addition, 7 of the 8 questions were subjected to the Round One consensus criteria. Group A reached consensus on all 7 questions. Group B reached consensus on only 5 questions.

Questions 6 and 7 asked whether programming identified either functional or technical requirements for design, respectively (Tables 5 and 6). Clearly, both Group A (100%) and Group B (96%) supported identifying functional requirements during programming. However, only Group A (86%) supported the inclusion of technical requirements in

TABLE 5

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 6

Facility programming identifies the functional building requirements for design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	20	95	13	43
(4) AGREE	1	5	16	53
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	0	0	0	0
(1) STRONGLY DISAGREE	0	0	1	3
SAMPLE SIZE	21		30	
MEAN	4.952		4.333	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.218		0.802	
CONSENSUS	YES		YES	

TABLE 6

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 7

Facility programming identifies the technical building requirements for design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	10	45	3	10
(4) AGREE	9	41	8	28
(3) UNDECIDED	2	9	2	7
(2) DISAGREE	1	5	14	48
(1) STRONGLY DISAGREE	0	0	2	7
SAMPLE SIZE	22		29	
MEAN	4.273		2.862	
MEDIAN	4.000		2.000	
STANDARD DEVIATION	0.827		1.217	
CONSENSUS	YES		NO	

programming. Group B did not reach a consensus on Question 7 with 38 percent "agreeing" and 55 percent "disagreeing" with the statement.

Questions 32 and 33 asked if either the quantitative or qualitative requirements of the client's (user/using agency's) organization should be included in the facility programming document, respectively (Tables 7 and 8). Both Group A (96%) and Group B (87%) supported including quantitative requirements. Group A (100%) strongly concurred that qualitative requirements should be included. However, Group B did not reach consensus on the statement. In Group B, 64 percent "agreed", 19 percent were "undecided", and 16 percent "disagreed" with Question 33.

TABLE 7

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 32

A facility programming document should include the quantitative requirements of the client's (user/using agency's) organization.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	16	73	8	26
(4) AGREE	5	23	19	61
(3) UNDECIDED	1	4	3	10
(2) DISAGREE	0	0	1	3
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.682		4.097	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.568		0.700	
CONSENSUS	YES		YES	

Question 38 asked if uncovering the true needs of the client (user/using agency) is a recurring problem (Table 9). Both Group A and Group B reached consensus on this issue. Group A (86%) and Group B (97%) "agreed" with the statement.

The type or detail of the information provided by programming is another issue. Question 49 asked if programming included: (1) details for contract documents production, (2) major issues for conceptual design, or (3) both details and issues (Table 10). Both Group A (100%) and Group B (93%) concurred that programming included major issues for conceptual design. However, only 36 percent and 32 percent from Group A and Group B, respectively, supported the inclusion of details for contract documents production.

TABLE 8

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 33

A facility programming document should include the qualitative requirements of the client's (user/using agency's) organization.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	17	77	5	16
(4) AGREE	5	23	15	48
(3) UNDECIDED	0	0	6	19
(2) DISAGREE	0	0	5	16
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.773		3.645	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.429		0.950	
CONSENSUS	YES		NO	

TABLE 9

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 38

During the programming process, uncovering the true needs of the client (user/using agency) is a recurring problem.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	12	54	18	58
(4) AGREE	7	32	12	39
(3) UNDECIDED	2	9	1	3
(2) DISAGREE	1	4	0	0
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.364		4.548	
MEDIAN	5.000		5.000	
STANDARD DEVIATION	0.848		0.568	
CONSENSUS	YES		YES	

TABLE 10

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 49

Programming includes:

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
Details for Contract Documents Production	0	0	2	7
Major Issues for Conceptual Design	14	64	19	68
Both	8	36	7	25
SAMPLE SIZE	22		28	
CONSENSUS	YES		YES	

Question 51 asked what type of information should almost always be included in a programming document (Table 11). The question listed 7 possible answers. Group A (100%) strongly endorsed including organizational goals and objectives in the programming document. In comparison, only 53 percent of Group B respondents supported programming documents containing organization goals. Both Group A (100%) and Group B (97%) strongly agreed that functional requirements should be incorporated in programming documents. This data supports the results from Question 6. However, Group A (68%) and Group B (60%) only moderately supported including technical requirements in programming documents. This data does not support the results from Question 7 in which 86 percent and 38 percent of Group A and Group B, respectively, "agreed" that programming identifies

TABLE 11

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 51

A programming document almost always should include:

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
Organizational Goals and Objectives	22	100	16	53
Functional Requirements	22	100	29	97
Technical Requirements	15	68	18	60
Budget and Cost Information	17	77	30	100
Schedule Information	13	59	18	60
Environmental Data	14	64	26	87
Energy Requirements	10	45	20	67
SAMPLE SIZE	22		30	

technical requirements. Another area examined was budget and cost information. Group B (100%) strongly endorsed including cost information. However, Group A (77%) only moderately supported the incorporating of cost information. Both Group A (59%) and Group B (60%) weakly supported including schedule information in programming documents. Group B (87%) strongly supported incorporating environmental data in programming. Group A (64%) only moderately promoted including environmental information. The final type of information asked about was energy requirements. Group B (67%) moderately supported including energy requirements. Less than half of Group A respondents (45%) endorsed adding energy criterion to programming information.

Programming Participants

In the programming process, there are various key players. These programming participants are: (1) the client, (2) the facility owner, (3) the facility user, (4) the designer, and (5) the programmer. In programming many of these parts are held by the same person or group. For example, the client, owner, and user may be the same person or group. The questionnaires contained 17 questions trying to define these players roles.

Questions 10 and 11 asked whether programming was the responsibility of the client/owner (user/using agency) or the designer, respectively (Tables 12 and 13). Consensus was not reached on either question. Group A (67%) showed a

TABLE 12

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 10

Programming is the responsibility of the client/owner (user/using agency).

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	11	50	2	6
(4) AGREE	4	18	10	32
(3) UNDECIDED	1	4	2	6
(2) DISAGREE	3	14	13	42
(1) STRONGLY DISAGREE	3	14	4	13
SAMPLE SIZE	22		31	
MEAN	3.773		2.774	
MEDIAN	4.500		2.000	
STANDARD DEVIATION	1.541		1.230	
CONSENSUS	NO		NO	

TABLE 13

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 11

Programming is the responsibility of the designer.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	5	24	1	3
(4) AGREE	5	24	6	19
(3) UNDECIDED	3	14	0	0
(2) DISAGREE	2	9	17	56
(1) STRONGLY DISAGREE	6	28	7	22
SAMPLE SIZE	21		31	
MEAN	3.048		2.258	
MEDIAN	3.000		2.000	
STANDARD DEVIATION	1.596		1.125	
CONSENSUS	NO		NO	

strong bias towards client/owner responsibility. Group B, however, did not hold either the user or the designer responsible. Group B (78%) did strongly lean towards "disagreeing" that programming was the designer's responsibility.

Question 48 is related to Questions 10 and 11. The question asked who should control the programming of facility projects (Table 14). The question was in multiple choice format and gave five possible responses. However, because Group A and Group B use different operating terms, the researcher could not directly compare the two groups' answers. Group A did not reach a consensus on who should control the programming process with responses split among the possible answers. Group B, however, did reach a

TABLE 14

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 48

In your opinion, who should control the programming of facility projects.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
Client/Owner	5	23	NA	
User/Using Agency	NA		0	0
Designer or Design Team	2	9	3	10
In-House Programming Staff (part of the design firm)	8	36	NA	
In-House Programming Staff	NA		24	83
Outside Programming Consultants (separate from the design firm)	3	14	NA	
Outside Programming Consultants (A-E Firms)	NA		0	0
Other	4	18	2	7
SAMPLE SIZE	22		29	
CONSENSUS	NO		YES	

consensus. Group B (83%) clearly thought that programming should be controlled by the in-house programming staff of Civil Engineering squadrons.

Questions 14, 16 and 39 deal with client (user/using agency) decision making during programming. Question 14 asked whether programming is a series of client design decisions (Table 15). Neither Group A nor Group B reached consensus on the question. Both groups were split among "agreeing" and "disagreeing" with the statement.

TABLE 15

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 14

Programming is a series of client design decisions.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	5	23	0	0
(4) AGREE	6	27	12	39
(3) UNDECIDED	0	0	3	10
(2) DISAGREE	6	27	11	35
(1) STRONGLY DISAGREE	5	23	5	16
SAMPLE SIZE	22		31	
MEAN	3.000		2.710	
MEDIAN	3.000		2.000	
STANDARD DEVIATION	1.574		1.160	
CONSENSUS	NO		NO	

In contrast, Question 16 asked whether a programmer should guide clients (users/using agencies) through decision making (Table 16). Group A (90%) and Group B (90%) both "agreed" with Question 16. Finally, Question 39 asked if getting clients (users/using agencies) to make decisions was a recurring problem. Again, Group A (82%) and Group B (80%) "agreed" with this statement.

Three questions (15, 17, and 19) requested information on the participation of the client/user and designer in programming. Question 15 asked whether client/user (user/using agency participation is very important in programming (Table 18). Group A (100%) and Group B (96%) "agreed" with the statement. Related to Question 15, Question 17 examined whether clients/users (users/using

TABLE 16

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 16

A programmer should guide clients (users/using agencies) through decision making.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	13	62	17	55
(4) AGREE	6	28	11	35
(3) UNDECIDED	0	0	2	6
(2) DISAGREE	2	10	1	3
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	21		31	
MEAN	4.429		4.419	
MEDIAN	5.000		5.000	
STANDARD DEVIATION	0.926		0.765	
CONSENSUS	YES		YES	

TABLE 17

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 39

During the programming process, getting clients (users/using agencies) to make decisions is a recurring problem.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	7	32	14	45
(4) AGREE	11	50	11	35
(3) UNDECIDED	3	14	1	3
(2) DISAGREE	1	4	5	16
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.091		4.097	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	0.811		1.076	
CONSENSUS	YES		YES	

TABLE 18

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 15

Client/user (user/using agency) participation is very important in programming.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	21	95	23	74
(4) AGREE	1	5	7	22
(3) UNDECIDED	0	0	1	3
(2) DISAGREE	0	0	0	0
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.955		4.710	
MEDIAN	5.000		5.000	
STANDARD DEVIATION	0.213		0.529	
CONSENSUS	YES		YES	

agencies) should be part of the programming team (Table 19). Again, Group A (96%) and Group B (93%) "agreed" with the statement. In comparison, Question 18 asked whether designers should be part of the programming team (Table 20). Neither group reached consensus on Question 18. However, both Group A (68%) and Group B (74%) demonstrated a bias towards "agreeing" with the statement.

Questions 19 and 20 asked if it was important to educate the client/users (users/using agencies) in the programming process and architectural design, respectively (Tables 21 and 22). Group A (96%) and Group B (87%) concurred that client/users need education in the programming process. In comparison, neither group reached

TABLE 19

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 17

Clients/users (users/using agencies) should be part of the programming team.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	16	73	20	64
(4) AGREE	5	23	9	29
(3) UNDECIDED	1	4	2	6
(2) DISAGREE	0	0	0	0
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.682		4.581	
MEDIAN	5.000		5.000	
STANDARD DEVIATION	0.568		0.620	
CONSENSUS	YES		YES	

TABLE 20

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 18

Designers should be part of the programming team.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	7	32	9	29
(4) AGREE	8	36	14	45
(3) UNDECIDED	3	14	5	16
(2) DISAGREE	2	9	3	10
(1) STRONGLY DISAGREE	2	9	0	0
SAMPLE SIZE	22		31	
MEAN	3.727		3.935	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.279		0.927	
CONSENSUS	NO		NO	

TABLE 21

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 19

It is important to educate the client/users (users/using agencies) in the programming process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	18	82	12	39
(4) AGREE	3	14	15	48
(3) UNDECIDED	1	4	4	13
(2) DISAGREE	0	0	0	0
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.773		4.258	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.528		0.682	
CONSENSUS	YES		YES	

TABLE 22

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 20

It is important to educate the client/users (users/using agencies) in architectural design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	7	32	1	3
(4) AGREE	10	45	9	29
(3) UNDECIDED	4	18	8	26
(2) DISAGREE	1	4	11	35
(1) STRONGLY DISAGREE	0	0	2	6
SAMPLE SIZE	22		31	
MEAN	4.045		2.871	
MEDIAN	4.000		3.000	
STANDARD DEVIATION	0.844		1.024	
CONSENSUS	NO		NO	

consensus on Question 20. However, Group A (77%) did present an inclination towards client/user education in architectural design. Group B was divided on the statement with 31 percent "agreeing" and 41 percent "disagreeing".

Questions 22 and 23 tried to determine if programming information is primarily for the designer or client (user/using agency), respectively (Tables 23 and 24). On Question 22, neither group reached a consensus. Both Group A and Group B were divided on the statement. In contrast, Group B (86%) did reach consensus on Question 23. Group B "disagreed" that programming information is primarily information for the client/user. However, Group A was divided on the same question.

TABLE 23

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 22

A facility programming document is primarily information for the designer.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	5	23	4	13
(4) AGREE	4	18	11	35
(3) UNDECIDED	3	14	3	10
(2) DISAGREE	9	41	9	29
(1) STRONGLY DISAGREE	1	4	4	13
SAMPLE SIZE	22		31	
MEAN	3.136		3.065	
MEDIAN	3.000		3.000	
STANDARD DEVIATION	1.320		1.316	
CONSENSUS	NO		NO	

TABLE 24

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 23

A facility programming document is primarily information for the client (user/using agency).

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	4	18	0	0
(4) AGREE	4	18	1	3
(3) UNDECIDED	3	14	3	10
(2) DISAGREE	9	41	25	83
(1) STRONGLY DISAGREE	2	9	1	3
SAMPLE SIZE	22		30	
MEAN	2.955		2.133	
MEDIAN	2.500		2.000	
STANDARD DEVIATION	1.327		0.213	
CONSENSUS	NO		YES	

In the literature, good communication was stated as a primary component of successful programming. Question 21 asked whether three-way communication between the designer, programmer, and client (user/using agency) is essential to programming (Table 25). Group A (82%) endorsed the statement. Group B did not reach a consensus. However, Group B (74%) did show a bias towards "agreeing" that three-way communication is essential.

Three questions (35, 36, and 37) deal the programmer's knowledge and skills. Question 35 asked if a programmer should have experience in design (Table 26). Neither group reached a consensus on the statement. However, Group A (68%) and Group B (67%) did establish partiality towards

TABLE 25

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 21

Three-way communication between the designer, programmer, and client (user/using agency) is essential to programming.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	14	64	17	55
(4) AGREE	4	18	6	19
(3) UNDECIDED	1	4	3	10
(2) DISAGREE	2	9	5	16
(1) STRONGLY DISAGREE	1	4	0	0
SAMPLE SIZE	22		31	
MEAN	4.273		4.129	
MEDIAN	5.000		5.000	
STANDARD DEVIATION	1.202		1.147	
CONSENSUS	YES		NO	

TABLE 26

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 35

A programmer should have experience in design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	7	32	6	19
(4) AGREE	8	36	15	48
(3) UNDECIDED	5	23	5	16
(2) DISAGREE	2	9	4	13
(1) STRONGLY DISAGREE	0	0	1	3
SAMPLE SIZE	22		31	
MEAN	4.045		3.677	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.133		1.045	
CONSENSUS	NO		NO	

"agreeing" a programmer should have design experience. Relating back to Question 21 on communication, Question 36 stated a programmer should be competent in communication skills, including graphic analysis and display (Table 27). Both Group A (100%) and Group B (100%) "agreed" with the statement. Finally, Question 37 inquired whether a programmer should understand the whole building process (Table 28). Group B (97%) "agreed" with the statement. However, Group A did not reach consensus with only 73 percent "agreeing" with the statement.

Programming and Design

A main focus of the research was the relationship between programming and design. In other words, how is programming information transformed into a design solution. The bulk of the questions deal with these two components of the facility delivery process. The following is the analysis of the 22 questions dealing with programming and design.

Questions 8 and 9 look broadly at what is programming and design, respectively. Question 8 asked if a facility programming document is a problem definition or statement (Table 28). Group A (100%) strongly supported this definition of a programming document. Group B did not reach a consensus on the same question. However, Group B (74%) did show a bias towards "agreement" with the statement. Similarly, Question 9 inquired whether a facility design

TABLE 27

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 36

A programmer should be competent in communication skills, including graphic analysis and display.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	17	77	14	45
(4) AGREE	5	23	17	55
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	0	0	0	0
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	4.773		4.452	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.429		0.506	
CONSENSUS	YES		YES	

TABLE 28

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 37

A programmer should understand the whole building delivery process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	10	45	12	39
(4) AGREE	6	27	18	82
(3) UNDECIDED	4	18	1	3
(2) DISAGREE	1	4	0	0
(1) STRONGLY DISAGREE	1	4	0	0
SAMPLE SIZE	22		31	
MEAN	4.045		4.355	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.133		0.551	
CONSENSUS	NO		YES	

TABLE 29

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 8

A facility programming document is a problem definition or statement.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	18	90	4	13
(4) AGREE	2	10	18	60
(3) UNDECIDED	0	0	1	3
(2) DISAGREE	0	0	5	17
(1) STRONGLY DISAGREE	0	0	2	7
SAMPLE SIZE	20		30	
MEAN	4.900		3.567	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.308		1.135	
CONSENSUS	YES		NO	

is a problem solution (Table 30). In contrast, Group B (94%) strongly supported this statement and Group A did not reach a consensus. Still Group A (65%) did demonstrate an inclination towards "agreeing" with Question 9.

Questions 12 and 13 asked if programming and design are iterative processes, respectively (Tables 31 and 32). Group A (91%) "agreed" that programming is an iterative process. Group B did not attain consensus on Question 12 with 55 percent "agreeing" and 29 percent "disagreeing" with the statement. Similarly, Group A (96%) "agreed" that design is an iterative process and Group B did not reach consensus. Of the Group B respondents, 57 percent "agreed" and 30 percent "disagreed" with Question 13.

TABLE 30

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 9

A facility design is a problem solution.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	11	55	8	27
(4) AGREE	2	10	20	67
(3) UNDECIDED	1	5	1	3
(2) DISAGREE	1	5	1	3
(1) STRONGLY DISAGREE	5	25	0	0
SAMPLE SIZE	20		30	
MEAN	3.650		4.167	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	1.755		0.648	
CONSENSUS	NO		YES	

TABLE 31

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 12

Programming is an iterative process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	15	68	5	16
(4) AGREE	5	23	12	39
(3) UNDECIDED	2	9	5	16
(2) DISAGREE	0	0	8	26
(1) STRONGLY DISAGREE	0	0	1	3
SAMPLE SIZE	22		31	
MEAN	4.500		3.387	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.913		1.145	
CONSENSUS	YES		NO	

TABLE 32

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 13

Design is an iterative process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	14	64	3	10
(4) AGREE	7	32	14	47
(3) UNDECIDED	1	4	4	13
(2) DISAGREE	0	0	7	23
(1) STRONGLY DISAGREE	0	0	2	7
SAMPLE SIZE	22		30	
MEAN	4.500		3.300	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.913		1.149	
CONSENSUS	YES		NO	

Question 41 is related to Question 12. The question asked how many opportunities, on the average, do your clients (users/using agencies) have to review, verify, change or add to the programming information (Table 33). The question was in multiple choice format. The statistics show Group A, as a whole, presented the client with close to 4 (3.810) occasions to review programming data. In comparison, Group B allowed the users nearly 3 (2.833) opportunities.

Like Question 41, Question 42 inquired how many design solutions, on the average, do you or your firm (A-E firm) present the client/owner (user/using agency) (Table 34). The results were that Group A submitted 3 solutions and Group B submitted about 2 (2.276) solutions. Relating back

TABLE 33

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 41

How many opportunities, on the average, do your clients (users/using agencies) have to review, verify, change or add to the programming information?

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(1) ONE	0	0	2	7
(2) TWO	2	9	10	33
(3) THREE	8	38	13	43
(4) FOUR	3	14	1	3
(5) FIVE OR MORE	8	38	4	13
SAMPLE SIZE	21		30	
MEAN	3.810		2.833	
MEDIAN	4.000		3.000	
STANDARD DEVIATION	1.078		1.085	

TABLE 34

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 42

How many design solutions, on the average, do you or your firm (A-E firm) present the client/owner (user/using agency)?

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(1) ONE	1	5	4	14
(2) TWO	2	10	15	52
(3) THREE	14	74	8	27
(4) FOUR	0	0	2	7
(5) FIVE OR MORE	2	5	0	0
SAMPLE SIZE	19		29	
MEAN	3.000		2.276	
MEDIAN	3.000		2.000	
STANDARD DEVIATION	0.882		0.797	

to Questions 12 and 13, Questions 41 and 42 would appear to support that programming and design are iterative processes.

An additional four questions (29, 30, 34 and 43) dealt exclusively with programming. Question 29 asked whether the programming process is the same for all facility projects (Table 35). Neither group reached consensus on the question. However, a majority of Group A (73%) and Group B (74%) respondents "disagreed" with the statement. Question 30 examined if programming is essential regardless of project size (Table 36). Clearly, Group A (96%) "agreed" with the statement. Group B, though, did not reach a consensus. However, 70 percent of the Group B respondents did "agree" with Question 30. Along the same lines,

TABLE 35

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 29

The programming process is the same for all facility projects.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	2	9	3	10
(4) AGREE	3	14	5	16
(3) UNDECIDED	1	4	0	0
(2) DISAGREE	9	41	18	58
(1) STRONGLY DISAGREE	7	32	5	16
SAMPLE SIZE	22		31	
MEAN	2.273		2.452	
MEDIAN	2.000		2.000	
STANDARD DEVIATION	1.316		1.234	
CONSENSUS	NO		NO	

TABLE 36

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 30

Programming is essential regardless of project size.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	16	73	7	22
(4) AGREE	5	23	15	48
(3) UNDECIDED	1	4	1	3
(2) DISAGREE	0	0	5	16
(1) STRONGLY DISAGREE	0	0	3	10
SAMPLE SIZE	22		31	
MEAN	4.636		3.581	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.727		1.285	
CONSENSUS	YES		NO	

Question 34 asked if programming should always produce an formal document (Table 37). Again, Group A (81%) "agreed" with the statement. Only 58 percent of the Group B participants responded favorably to the same question. Finally, Question 43 asked the respondents what percentage of overall project development time should be spent on programming (Table 38). A majority of Group A (60%) and Group B (62%) indicated that programming should require 5 to 15 percent of project development time.

In the next four questions, the researcher tried to establish how the two groups view design in the facility delivery process. Question 47 asked the respondents what are the distinct phases of the facility delivery process (Table 39). Neither group achieved consensus on the

TABLE 37

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 34

Programming should always produce an formal document.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	10	45	2	6
(4) AGREE	8	36	16	52
(3) UNDECIDED	2	9	2	6
(2) DISAGREE	2	9	8	26
(1) STRONGLY DISAGREE	0	0	3	10
SAMPLE SIZE	22		31	
MEAN	4.182		3.194	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	0.958		1.195	
CONSENSUS	YES		NO	

TABLE 38

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 43

In your opinion, what percentage of overall project development time should be spent on programming.

	GROUP A		GROUP B	
PERCENTAGE	FREQ	PERC	FREQ	PERC
LESS THAN 5%	3	15	3	10
5% TO 10%	8	40	11	38
11% TO 15%	4	20	7	24
16% TO 20%	3	15	2	7
21% TO 25%	2	10	4	14
26% OR MORE	0	0	2	7
SAMPLE SIZE	20		29	

TABLE 39

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 47

The distinct phases of the facility delivery process are:

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
PROGRAMMING, CONCEPTUAL DESIGN, DESIGN AND CONSTRUCTION	13	59	17	55
PROGRAMMING, DESIGN AND CONSTRUCTION	2	9	12	39
DESIGN AND CONSTRUCTION	1	4	1	3
OTHER	6	27	1	3
SAMPLE SIZE	22		31	
CONSENSUS	NO		NO	

question. Though, a majority of Group A (59%) and Group B (55%) answered that the specific phases were: (1) programming, (2) conceptual design, (3) design (contract documents production), and (4) construction. Examining the design portion of the facility delivery process, Question 24 inquired whether conceptual design and contract documents production are two separate phases of the design process (Table 40). Both Group A (82%) and Group B (84%) "agreed" with the statement. However, the results of Question 24 do not appear to support the responses from Question 47. Questions 25 and 45 looked more closely at where conceptual design fits into the facility delivery process. Question 25 asked if conceptual design is part of the programming process (Table 41). Neither group reached consensus on the

TABLE 40

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 24

Conceptual design and contract documents production are two separate phases of the design process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	12	55	9	29
(4) AGREE	6	27	17	55
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	4	18	4	13
(1) STRONGLY DISAGREE	0	0	1	3
SAMPLE SIZE	22		31	
MEAN	4.182		3.935	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	1.140		1.063	
CONSENSUS	YES		YES	

TABLE 41

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 25

Conceptual design is part of the programming process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	4	18	4	13
(4) AGREE	3	14	14	47
(3) UNDECIDED	4	18	6	20
(2) DISAGREE	8	36	6	20
(1) STRONGLY DISAGREE	3	14	0	0
SAMPLE SIZE	22		30	
MEAN	2.864		3.533	
MEDIAN	2.500		4.000	
STANDARD DEVIATION	1.356		0.973	
CONSENSUS	NO		NO	

statement. Group A participants responded with 32 percent "agreeing", 18 percent "undecided", and 50 percent "disagreeing" on the question. In contrast, Group B answered with 60 percent "agreeing", 20 percent "undecided", and 20 percent "disagreeing". Question 46 asked a similar question in multiple choice format. The question was conceptual design is: (A) part of the programming process, (B) part of the design process, (C) part of both the programming and design processes, or (D) separate from the programming and design processes (Table 42). Group B reached a consensus with 70 percent of the participants responding that conceptual design was part of both the

TABLE 42

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 46

Conceptual Design is:

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
PART OF THE PROGRAMMING PROCESS	4	18	4	13
PART OF THE DESIGN PROCESS	10	45	5	17
PART OF BOTH THE PROGRAMMING AND DESIGN PROCESSES	6	27	21	70
SEPARATE FROM THE PROGRAMMING AND DESIGN PROCESSES	2	9	0	0
SAMPLE SIZE	22		30	
CONSENSUS	NO		YES	

programming and design processes. Group A, however, split their responses among the four possible answers.

Seven questions inquire about programming's relationship to design. The Architect's Guide to Facility Programming describes three basic approaches to programming. Question 50 asked which approach best described the respondent's programming method (Table 43). The question was given in multiple choice format defining each method as follows:

1. Segregated. Programming is a separate distinct activity performed prior to initiating design, and performed by separate individuals or teams from the designers.

2. Integrated. Programming is not a "predesign" service, but an integral first part of the design process.

3. Interactive. Programming and designing are performed in alternating sequence and in response to each other.

Group A and Group B did not reach a consensus on the question. The responses for the two groups were divided among the possible answers. However, in both Group A (41%) and Group B (52%), the segregated method was the most frequent response.

Questions 26, 27 and 28 were related to each of three approaches listed in Question 50. Question 26 asked if programming should be completed prior to design (Table 44). Group B (86%) "agreed" with the statement. However, Group A

TABLE 43

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 50

The Architect's Guide to Facility Programming lists three basic approaches to programming, which of the following approaches best describes your programming method.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
SEGREGATED	9	41	16	52
INTEGRATED	6	27	4	13
INTERACTIVE	5	23	11	35
SEGREGATED OR INTERACTIVE	2	9	0	0
SAMPLE SIZE	22		31	
CONSENSUS	NO		NO	

TABLE 44

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 26

Programming should be completed prior to design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	9	41	13	43
(4) AGREE	4	18	13	43
(3) UNDECIDED	3	14	0	0
(2) DISAGREE	4	18	3	10
(1) STRONGLY DISAGREE	2	9	1	3
SAMPLE SIZE	22		30	
MEAN	3.636		4.133	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.432		1.074	
CONSENSUS	NO		YES	

did not reach a consensus with 59 percent "agreeing", 14 percent "undecided", and 27 percent "disagreeing" on the question. Question 27 inquired whether programming should be integrated with design (Table 45). Neither group achieved consensus on the question. Answers were split among the possible responses with no majority. Question 28 asked whether programming and design should be interactive, not separate phases of the facility delivery process (Table 46). Again, the two groups did not reach consensus. However, a majority of Group A (59%) and Group B (58%) respondents "agreed" with the statement.

The last three questions in this section, continue to examine programming's relationship with design. Question 31

TABLE 45

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 27

Programming should be integrated with design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	3	14	3	10
(4) AGREE	7	32	8	26
(3) UNDECIDED	4	18	7	22
(2) DISAGREE	7	32	13	42
(1) STRONGLY DISAGREE	1	4	0	0
SAMPLE SIZE	22		31	
MEAN	3.182		3.032	
MEDIAN	3.000		3.000	
STANDARD DEVIATION	1.181		1.048	
CONSENSUS	NO		NO	

TABLE 46

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 28

Programming and design should be interactive, not separate phases of the facility delivery process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	5	23	4	13
(4) AGREE	8	36	14	45
(3) UNDECIDED	3	14	5	16
(2) DISAGREE	5	23	8	26
(1) STRONGLY DISAGREE	1	4	0	0
SAMPLE SIZE	22		31	
MEAN	3.500		3.452	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.225		1.028	
CONSENSUS	NO		NO	

stated that the end product of programming is information, not design (Table 47). Though neither group reached consensus, a clear majority of Group A (71%) and Group B (77%) "agreed" on the question. Question 40 asked whether the programming - design relationship/connection is a recurring problem (Table 48). Once more, neither group achieved consensus. A majority of Group A (59%), however, did "agree" with the statement. Group B, though, was divided with 41 percent "agreeing" and 45 percent "disagreeing". Finally, Question 45 asked if programming was either part of or separate from the design process (Table 49). Neither Group A or B achieved consensus with both evenly divided among the possible responses.

TABLE 47

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 31

The end product of programming is information not design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	12	57	9	29
(4) AGREE	3	14	15	48
(3) UNDECIDED	2	9	3	10
(2) DISAGREE	3	14	4	13
(1) STRONGLY DISAGREE	1	5	0	0
SAMPLE SIZE	21		31	
MEAN	4.048		3.935	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	1.322		0.964	
CONSENSUS	NO		NO	

TABLE 48

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 40

During the facility delivery process, the programming - design relationship/connection is a recurring problem.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	1	4	2	6
(4) AGREE	12	55	11	35
(3) UNDECIDED	2	9	4	13
(2) DISAGREE	7	32	14	45
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		31	
MEAN	3.318		3.032	
MEDIAN	4.000		3.000	
STANDARD DEVIATION	0.995		1.048	
CONSENSUS	NO		NO	

TABLE 49

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 45

Programming is ___________ the aesign process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
PART OF	10	45	13	50
SEPARATE FROM	9	41	13	50
INTERACTIVE WITH	3	14	0	0
SAMPLE SIZE	22		26	
CONSENSUS	NO		NO	

Programming Techniques

The research, also, examined the types of progr mming techniques used by the two respondent groups. Programming techniques collect, analyze, organize, evaluate and present information. Questions were written to determine which techniques were most widely used by each group.

Question 52 asked which techniques had the respondents used to collect programming information (Table 50). The list of answers included 17 techniques falling into three broad categories: (1) research or background methods, (2) observation methods, and (3) comparison methods. 50 percent or more of the Group A respondents had used 9 of 17 the techniques. 6 of the 9 techniques fell into the research and background methods category. 2 of the 9 techniques were observation methods. Only 1 of 9 was a comparison method. In comparison, 50 percent or more of the Group B

TABLE 50

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 52

Which of the following techniques have you used when collecting programming information.

	GROUP A		GROUP B	
TECHNIQUE	FREQ	PERC	FREQ	PERC
Interviews	22	100	28	95
Direct Observation	22	100	21	72
Background Data Research	20	91	23	79
Surveys	22	100	20	69
Questionnaires	22	100	9	31
Participant Observation	14	64	16	55
Standardized Data Forms	17	77	10	34
Data Logs	12	55	9	31
Preference Matrix	14	64	3	10
Ranking Chart	10	45	3	10
Instrumented Observation	3	14	6	21
Tracking	4	18	4	14
Behavior Mapping	7	32	1	3
Adjective Checklist	7	32	1	3
Semantic Differential	6	27	0	0
Attribute Discrimination Scale	2	9	0	0
Behavior Specimen Record	1	4	0	0
SAMPLE SIZE	22		29	

respondents had used only 5 different techniques. 3 of the 5 were research and background techniques. The remaining 2 of 5 were observation methods.

Question 53 requested which techniques the respondents had used for analyzing and organizing programming data (Table 51). The list of possible answers included 35

TABLE 51

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 53

Which of the following techniques have you used when analyzing and organizing programming information.

	GROUP A		GROUP B	
TECHNIQUE	FREQ	PERC	FREQ	PERC
Project Cost Estimating	16	73	26	96
Bubble Diagram	21	95	18	67
Construction Cost Estimating	15	68	21	78
Life Cycle Cost Analysis	13	59	23	85
Functional Relationship Diagram	20	91	15	55
Flow Diagram	19	86	13	48
Organizational Chart	20	91	12	44
Space Unit Standards	21	95	11	41
Space Program	22	100	9	33
Cost-Benefit Analysis	12	55	18	67
Layout Diagram	14	64	15	55
Descriptive Statistics	18	82	8	30
Bar Chart/Milestone Chart	18	82	8	30
Relationship Matrices	19	86	6	22
Worksheets	11	50	13	48
Adjacency Diagram	21	95	1	4
Energy Budgeting	10	45	11	41
Critical Path Method	12	55	8	30
Block Diagram	14	64	4	15
Activity Time Chart	12	55	3	11
Activity Site Model	7	32	7	26
Inferential Statistics	12	55	1	4
Program Evaluation and Review Tech.	8	36	5	18
Value Analysis	7	32	6	22
Time Budget Analysis	6	27	4	15
Interaction Net	7	32	3	11
Behavior Map	5	23	1	4
Link-Mode Diagram	4	18	2	7
Pattern Language	6	27	0	0
Analysis Cards	5	23	0	0
SAMPLE SIZE	22		27	

techniques in 7 main categories: (1) statistical analysis, (2) functional and activity analysis, (3) space analysis, (4) cost analysis, (5) scheduling, (6) relationship matrices, and (7) correlation diagrams. 50 percent or more of the Group A respondents indicated use of 20 of the 35 techniques. However, 50 percent of more of the Group B respondents only responded to 7 of the techniques. For Group A, 2 were statistical analysis techniques, 2 were space analysis techniques, 4 were cost analysis techniques, 3 were scheduling techniques, and 7 were correlation diagrams. The 2 remaining techniques were relationship matrices and worksheets, categories in themselves. In comparison, for Group B, 4 were cost analysis techniques and 3 were correlation diagrams.

Question 54 asked which techniques the respondents had used for communicating and evaluating programming data (Table 52). The possible answers were a list 18 techniques in 3 categories: (1) collective decision making methods, (2) presentation and documentation methods, and (3) rating methods. 50 percent or more of the Group A respondents indicated use of 11 of the 18 methods. Of these 11, 2 were collective decision techniques, 6 were presentation and documentation techniques, and 2 were rating techniques. In comparison, 50 percent or more of the Group B respondents responded to only 5 different techniques. Of the 5, 2 were collective decision techniques and 3 were presentation and documentation techniques.

TABLE 52

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 54

Which of the following techniques have you used when communicating and evaluating programming information.

	GROUP A		GROUP B	
TECHNIQUE	FREQ	PERC	FREQ	PERC
Graphics	22	100	23	79
Oral Presentations	21	95	24	83
Narrative	19	86	24	83
Brainstorming	15	68	25	86
Group Planning	13	59	20	69
Audio/Visual Aids	19	86	13	45
Panel Discussions	7	32	14	48
Forums	11	50	8	27
Evaluation Matrix	12	55	5	17
Buzz/Rap Session	3	14	13	45
Weighting	11	50	5	17
Work/Charrette/ Primer Books	11	50	3	10
Rating and Rating Scales	12	55	1	3
Gaming	8	36	3	10
Rating Chart	8	36	1	3
Role Playing	2	9	2	7
Synetics	2	9	2	7
Ladder Scale	1	4	0	0
SAMPLE SIZE	22		29	

In a related question to programming techniques, Question 44 asked how much do the respondents use a computer to perform the analyzing, organizing and evaluation or programming data (Table 53). For Group A, 100 percent indicated using the computer to do most of some of the data processing. However, only 61 percent of Group B indicated the same amount of computer use.

TABLE 53

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 44

You or your firm use a computer (not including word processing) to perform __________ of the analyzing, organizing and evaluating of programming data.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
MOST	7	32	9	29
SOME	15	68	10	32
LITTLE	0	0	8	26
NONE	0	0	4	13
SAMPLE SIZE	22		31	

Open-Ended Questions

Both Group A and Group B were asked one unstructured question requiring a written response. The two questions were different for each respondent group.

Group A was asked "What two or three questions would you like to ask your peers about facility programming?" The intent of the question was to uncover any prominent areas of concern among the professional programmers. Of the 22 participants, 12 answered with 24 separate questions. No one question or area of concern appeared as dominant. However, the questions seemed to fall into broad two categories: (1) professional practice, and (2) programming methods and techniques. Under professional practice, questions inquired about:

1. Fee structures.

2. Procurement requirements.

3. Professional licensing.

4. Professional liability.

5. Marketing.

Other questions dealt mainly with information gathering or resources, and client communications.

Group B was asked "Do you believe the Air Force programming methods adequately define project requirements prior to initiating design?" (Table 54). In addition, the respondents were requested to explain their answers. Of the 31 participants, 28 answered the question. The results indicated that 57 percent did not believe the Air Force methods were adequate. However, the researcher used his judgment to categorize an answer if a clear yes or no response was not received. In addition, of the 12 respondents who answered "yes", 8 indicated potential

TABLE 54

ROUND ONE DESCRIPTIVE STATISTICS FOR QUESTION 55
(GROUP B ONLY)

Do you believe Air Force programming methods adequately define project requirements prior to initiating design?

RESPONSE	FREQ	PERC
YES	12	43
NO	16	57
SAMPLE SIZE	28	

areas of improvement in the programming process. The most frequent reasons given for inadequate programming were:

1. Personnel changes.
2. Amount of time between programming and design.
3. Workload.
4. Cost limitations.
5. Programmers' lack of experience.

Personnel changes in the using agency, especially with commanders, was clearly the reason given most often for programming problems. Closely related to personnel changes was the "lag time" between programming and design. Within the elapsed time in process, personnel change bringing different personal attitudes or values into the project.

Chapter Summary

Chapter IV summarized the results of the Round One Delphi Questionnaire. The data included the descriptive statistics on each question including: (1) response frequencies, (2) response percentages, (3) the mean, (4) the median, (5) the standard deviation, and (6) sample size. In addition, the first round data was used to determine consensus on a particular question.

Chapter V examines the results of the Round Two Delphi Questionnaire. The questions that did not meet the round one consensus criteria were included in the second round. They represented areas of disagreement among the participant groups and required further examination.

V. Round Two Delphi Questionnaire Results

Chapter Overview

This chapter reports the results of the second round questionnaires for the two research groups. Questions that did not meet the first round consensus criteria are included in round two. The Group A and Group B survey instruments 25 and 26 questions, respectively. The resulting data is broken into 4 broad categories for review: (1) Programming Content, (2) Programming Participants, (3) Programming and Design Interaction, and (4) Programming Techniques. The chapter narrative is accompanied by the statistical data presented in tabular form comparing the two groups of respondents.

General Results

The researcher's goal was a total of 20 participants in each of the respondent groups. This objective was achieved in the second round.

Group A. 20 of the 25 questionnaires were completed and returned over a six week period. The response rate was 80 percent. Consensus was reached on 15 of the 22 applicable questions in the second round.

Group B. 20 of the 31 questionnaires were completed and returned over a six week period. The response rate was 64.5 percent. Consensus was reached on 14 of the 23 applicable questions.

Criteria for Consensus

The main objective of the research method, the Delphi Technique, is the consensus of respondents on an issue or question. For the purposes of the Round Two questionnaires, the criteria for consensus for multiple choice and rated scale questions was:

Multiple Choice. A 60 percent agreement among respondents on a single answer, multiple choice question constituted consensus.

Rated Scale. A 70 percent agreement among respondents on rated scale questions constituted consensus based on two groups of responses: "strongly agree/agree" and "strongly disagree/disagree."

The Group A and Group B survey instruments contained many of the same questions. However, when a question was included in only one of the questionnaires, the other groups' first round data was included for comparison.

Programming Content

Of the 7 round one questions on programming content, 2 were included in Group B's second round questionnaire. The 2 questions were 7 and 33. Group A had reached consensus on all applicable questions, so none were repeated in their round two survey.

Question 7 asked whether programming identified the technical building requirements for design (Table 55). The original question was not altered, but a definition for the

TABLE 55

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 7

Facility programming identifies the technical building requirements for design.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	10	45	1	5
(4) AGREE	9	41	7	35
(3) UNDECIDED	2	9	0	0
(2) DISAGREE	1	5	12	60
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		20	
MEAN	4.273		2.850	
MEDIAN	4.000		2.000	
STANDARD DEVIATION	0.827		1.089	
CONSENSUS	YES		NO	

* Data from Rouıd One Questionnaire

word "requirements" was added to clarify its meaning. Group B, however, did not reach consensus on the question with 40 percent "agreeing" and 60 percent "disagreeing" with the statement.

Question 33 asked whether the qualitative requirements of the user/using agency's organization should be included in the facility programming document (Table 56). The original question was not altered. However, definitions for two key phrases, "facility programming document" and "qualitative requirements" were include the clarify the question. Group B (85%) supported the including qualitative requirements in the programming document.

TABLE 56

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 33

A facility programming document should include the qualitative requirements of the client's (user/using agency's) organization.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	17	77	2	10
(4) AGREE	5	23	15	75
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	0	0	3	15
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		20	
MEAN	4.773		3.800	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.429		0.833	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

Programming Participants

Of the original 17 questions on the roles of key players in programming, 11 were included in either the Group A or Group B round two questionnaire. For Group A, 10 questions about programming participants were contained in the survey instrument. For Group B, 9 questions were incorporated in their round two questionnaire.

Questions 10 and 11 asked whether programming was the responsibility of client/owner (user/using agency) or the designer, respectively (Tables 57 and 58). The questions were not altered from the round one questionnaire. However, a definition for "responsibility" was contained in the

TABLE 57

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 10

Programming is the responsibility of the client/owner (user/using agency).

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	8	40	1	5
(4) AGREE	6	30	4	20
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	5	25	15	75
(1) STRONGLY DISAGREE	1	5	0	0
SAMPLE SIZE	20		20	
MEAN	3.750		2.550	
MEDIAN	4.000		2.000	
STANDARD DEVIATION	1.372		0.999	
CONSENSUS	YES		YES	

TABLE 58

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 11

Programming is the responsibility of the designer.

	GROUP A		GROUP B*	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	2	10	1	3
(4) AGREE	6	30	6	19
(3) UNDECIDED	1	5	0	0
(2) DISAGREE	4	20	17	56
(1) STRONGLY DISAGREE	7	35	7	22
SAMPLE SIZE	20		31	
MEAN	2.600		2.258	
MEDIAN	2.000		2.000	
STANDARD DEVIATION	1.501		1.125	
CONSENSUS	NO		YES	

* Data from Round One Questionnaire

survey instrument. Both Group A and Group B reached consensus on Question 10. Group A (70%) "agreed" that programming was a client responsibility. In contrast, Group B (75%) "disagreed" that user/using agency was responsible for programming. Question 11 was included only in Group A's second round questionnaire. Group A, however, did not reach consensus with 40 percent "agreeing" and 55 percent "disagreeing" with designer responsibility for programming.

Question 48 requested information on who should control the programming of facility projects (Table 59). This question was contained only in the Group A questionnaire. The question, however, was altered. First, the respondents were told to assume the client/owner had no in-house programming capability and that the design firm had an in-house programming staff. Second, the phrase "programming process" replaced "programming" in the question. In addition, the questionnaire contained a definition for the word "control" to clarify the question. With the clarifications and changes, Group A reached consensus with 63 percent saying the design firm's in-house programming staff should control the programming process.

Question 14 asked if programming is a series of client (user/using agency) decisions on the direction of design (Table 60). The original question was changed by replacing "design decisions" with "decisions on the direction of design." Neither group reached consensus on the question, but both groups showed a strong bias towards "agreement."

TABLE 59

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 48

In your opinion, who should control the programming of facility projects.

	GROUP A		GROUP B*	
	FREQ	PERC	FREQ	PERC
Client/Owner	2	11	NA	
User/Using Agency	NA		0	0
Designer or Design Team	0	0	3	10
In-House Programming Staff (part of the design firm)	12	63	NA	
In-House Programming Staff	NA		24	83
Outside Programming Consultants (separate from the design firm)	2	11	NA	
Outside Programming Consultants (A-E Firms)	NA		0	0
Other	3	16	2	7
SAMPLE SIZE	19		29	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

Question 18 was repeated for both groups in the round two questionnaires. The question asked whether designers should be part of the programming team (Table 61). Both Group A (75%) and Group B (70%) "agreed" with the statement.

Question 20 asked if it was important to educate the client/users (users/using agency) in the architectural design process (Table 62). The original question was

TABLE 60

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 14

Programming is a series of client decisions on the direction of design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	4	20	0	0
(4) AGREE	8	40	13	65
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	4	20	7	35
(1) STRONGLY DISAGREE	4	20	0	0
SAMPLE SIZE	20		20	
MEAN	3.200		3.300	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.508		0.979	
CONSENSUS	NO		NO	

TABLE 61

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 18

Designers should be part of the programming team.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	6	30	2	10
(4) AGREE	9	45	12	60
(3) UNDECIDED	2	10	2	10
(2) DISAGREE	3	15	4	20
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	20		20	
MEAN	3.900		3.600	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.021		0.940	
CONSENSUS	YES		YES	

TABLE 62

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 20

It is important to educate the client/users (users/using agencies) in architectural design process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	15	75	0	0
(4) AGREE	4	20	11	55
(3) UNDECIDED	1	5	3	15
(2) DISAGREE	0	4	6	30
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	20		20	
MEAN	4.700		3.250	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.571		0.910	
CONSENSUS	YES		NO	

altered by replacing "architectural design" with "architectural design process." Group A (95%) "strongly agreed" with the statement. However, Group B did not reach consensus with 55 percent "agreeing" and 30 percent "disagreeing" with educating users/using agencies in the design process.

Questions 22 and 23 tried to determine who benefited from the programming information (Tables 63 and 64). Question 22 inquired whether a facility programming document was primarily information for the designer. Neither group reached consensus with both almost evenly split on the validity of the statement. In contrast, Question 23 asked whether a facility programming document was valuable

TABLE 63

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 22

A facility programming document is primarily information for the designer.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	3	15	1	5
(4) AGREE	7	35	10	50
(3) UNDECIDED	0	0	1	5
(2) DISAGREE	10	50	7	35
(1) STRONGLY DISAGREE	0	0	1	5
SAMPLE SIZE	20		20	
MEAN	3.150		3.150	
MEDIAN	3.000		4.000	
STANDARD DEVIATION	1.226		1.137	
CONSENSUS	NO		NO	

TABLE 64

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 23

A facility programming document is valuable information for the client (user/using agency).

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	10	50	0	0
(4) AGREE	1	5	6	30
(3) UNDECIDED	2	10	2	10
(2) DISAGREE	6	30	12	60
(1) STRONGLY DISAGREE	1	5	0	3
SAMPLE SIZE	20		20	
MEAN	3.650		2.700	
MEDIAN	4.500		2.000	
STANDARD DEVIATION	1.496		0.923	
CONSENSUS	NO		NO	

information for the client (user/using agency). The original question was altered by replacing "primarily" with "valuable." Even with the change neither group reached consensus. In comparison, though, Group A leaned towards "agreement", while Group B tended to "disagree" that the programming document contained valuable data for the user.

Only Group B was asked Question 21 in round two. The question inquired if three-way communication between the designer, programmer, and client (user/using agency) was essential to programming (Table 65). The Group B respondents reached consensus with 80 percent agreeing with the statement.

TABLE 65

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 21

Three-way communication between the designer, programmer, and client (user/using agency) is essential to programming.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	14	64	12	60
(4) AGREE	4	18	4	20
(3) UNDECIDED	1	4	1	5
(2) DISAGREE	2	9	3	15
(1) STRONGLY DISAGREE	1	4	0	0
SAMPLE SIZE	22		20	
MEAN	4.273		4.250	
MEDIAN	5.000		5.000	
STANDARD DEVIATION	1.202		1.118	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

Questions 35 and 37 dealt with the programmer's or programming team's knowledge and experience. Question 35 asked if a programmer of someone on the programming team should have experience in design (Table 66). Both Group A (95%) and Group B (75%) "agreed" with the statement. Question 37 inquired whether a programmer or someone on the programming team should understand the whole building delivery process (Table 67). The question was included only in round two Group A survey. Group A reached consensus with 80 percent "agreeing" with the statement. However, both Questions 35 and 37 were altered by adding the phrase "or someone on the programming team" in the second round.

TABLE 66

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 35

A programmer or someone on the programming team should have experience in design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	10	50	3	15
(4) AGREE	9	45	12	60
(3) UNDECIDED	1	5	2	10
(2) DISAGREE	0	0	3	15
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	20		20	
MEAN	4.450		3.750	
MEDIAN	4.500		4.000	
STANDARD DEVIATION	0.605		0.910	
CONSENSUS	YES		YES	

TABLE 67

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 37

A programmer or someone on the programming team should understand the whole building delivery process.

	GROUP A		GROUP B*	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	12	60	12	39
(4) AGREE	4	20	18	82
(3) UNDECIDED	4	20	1	3
(2) DISAGREE	0	0	0	0
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	20		31	
MEAN	4.400		4.355	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.821		0.551	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

Programming and Design

Of the original 22 questions dealing with programming and design, 17 were included in either the Group A or Group B round two questionnaires. For Group A, 12 of the 17 were contained in their survey instrument. For Group B, 14 of the 17 questions were included in their round two questionnaire.

Questions 8 and 9 looked generally at what is the end product of programming and design. Only Group B was asked Question 8. The question inquired whether a facility programming document is a problem definition or statement (Table 68). The wording of the question was not changed,

TABLE 68

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 8

A facility programming document is a problem definition or statement.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	18	90	2	10
(4) AGREE	2	10	17	85
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	0	0	1	5
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	20		20	
MEAN	4.900		4.000	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.308		0.562	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

but a definition of "facility programming document" was added to the questionnaire. In round two, Group B (95%) supported the statement. In comparison, only the Group A survey contained Question 9. This question asked if a facility design is a problem solution (Table 69). Group A (95%) strongly "agreed" with the statement in the second round.

Questions 12 and 13 asked if programming and design were iterative processes, respectively (Tables 70 and 71). Only the Group B questionnaires contained these questions. In addition, a definition for the word "iterative" was included in the survey instrument. Group B (75%) "agreed"

TABLE 69

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 9

A facility design is a problem solution.

	GROUP A		GROUP B*	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	17	85	8	27
(4) AGREE	2	10	20	67
(3) UNDECIDED	0	0	1	3
(2) DISAGREE	1	5	1	3
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	20		30	
MEAN	4.750		4.167	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.716		0.648	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

that programming is an iterative process. The Group B respondents, also, strongly supported with 90 percent saying design is an iterative process.

In round two, three questions (29, 30 and 34) dealt only with programming or the programming process. Question 29 asked if the programming process is the same for all facility projects (Table 72). Both Group A (80%) and Group B (90%) "disagreed" with this statement. Only the Group B's second round questionnaire contained Questions 30 and 34. Question 30 inquired whether programming is essential regardless of project size (Table 73). Group B supported the statement with 80 percent of the respondents "agreeing." Related to Question 30, Question 34 asked if programming

TABLE 70

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 12

Programming is an iterative process.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	15	68	2	10
(4) AGREE	5	23	13	65
(3) UNDECIDED	2	9	1	5
(2) DISAGREE	0	0	4	20
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		20	
MEAN	4.500		3.650	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.913		0.933	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

should always produce a formal document (Table 74). However, Group B did not reach consensus on this question with 57 percent "agreeing" and 42 percent "disagreeing" with the statement.

The remaining questions in this section deal with the programming - design relationship. To clarify the questions, definitions for "conceptual design" and "design" were contained in the questionnaires, as follows:

1. <u>Conceptual Design</u> means conceptual or schematic design per A.I.A standards.

2. <u>Design</u> means Design Development and Contract Documents Production per A.I.A. standards.

TABLE 71

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 13

Design is an iterative process.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	14	64	3	15
(4) AGREE	7	32	15	75
(3) UNDECIDED	1	4	0	0
(2) DISAGREE	0	0	2	10
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		20	
MEAN	4.500		3.950	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.913		0.760	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

TABLE 72

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 29

The programming process is the same for all facility projects.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	2	10	0	0
(4) AGREE	2	10	2	10
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	9	45	17	85
(1) STRONGLY DISAGREE	7	35	1	5
SAMPLE SIZE	20		20	
MEAN	2.150		2.150	
MEDIAN	2.000		2.000	
STANDARD DEVIATION	1.309		0.671	
CONSENSUS	YES		YES	

TABLE 73

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 30

Programming is essential regardless of project size.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	16	73	4	20
(4) AGREE	5	23	12	60
(3) UNDECIDED	1	4	0	0
(2) DISAGREE	0	0	4	20
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		20	
MEAN	4.636		3.800	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.727		1.005	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

With the next 3 questions, the researcher tried to establish how the two groups view design in the facility delivery process. Question 47 asked the respondents what are the distinct phases in of the facility delivery process (Table 75). Both Group A (78%) and Group B (85%) "agreed" that the specific phases were: (1) programming, (2) conceptual design, (3) design, and (4) construction.

Questions 25 and 45 examined more closely how conceptual design fits into the facility delivery process. Question 25 asked if conceptual design is part of the programming process (Table 76). Both groups reached consensus on the statement. Group A (70%) "disagreed" with the conceptual design is part of programming. In

TABLE 74

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 34

Programming should always produce an formal document.

	GROUP A*		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	10	45	2	10
(4) AGREE	8	36	9	47
(3) UNDECIDED	2	9	0	0
(2) DISAGREE	2	9	8	42
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	22		19	
MEAN	4.182		3.263	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	0.958		1.147	
CONSENSUS	YES		NO	

* Data from Round One Questionnaire

TABLE 75

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 47

The distinct phases of the facility delivery process are:

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
PROGRAMMING, CONCEPTUAL DESIGN, DESIGN AND CONSTRUCTION	14	78	17	85
PROGRAMMING, DESIGN AND CONSTRUCTION	2	11	3	15
DESIGN AND CONSTRUCTION	0	0	0	0
OTHER	2	11	0	0
SAMPLE SIZE	18		31	
CONSENSUS	YES		YES	

TABLE 76

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 25

Conceptual design is part of the programming process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	3	15	2	10
(4) AGREE	2	10	16	80
(3) UNDECIDED	1	5	2	10
(2) DISAGREE	11	55	0	0
(1) STRONGLY DISAGREE	3	15	0	0
SAMPLE SIZE	20		20	
MEAN	2.550		4.000	
MEDIAN	2.000		4.000	
STANDARD DEVIATION	1.317		0.459	
CONSENSUS	YES		YES	

contrast, Group B with 90 percent "agreeing" supported the statement. Question 46 asked a similar question in multiple choice format (Table 77). Only the Group A questionnaire included this question. Group A reached consensus with 72 percent of the respondents answering that conceptual design is part of the design process.

<u>The Architect's Guide to Facility Programming</u> describes three basic approached to programming. Originally, Question 50 asked which approach best described the respondent's programming method. For Group A, the question remained unchanged except for including the additional response choice of "segregated or interactive." For Group B, the question was reworded to asking which approach, in their opinion was best. The reason for the change is that Air

TABLE 77

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 46

Conceptual Design is:

	GROUP A		GROUP B*	
	FREQ	PERC	FREQ	PERC
PART OF THE PROGRAMMING PROCESS	1	6	4	13
PART OF THE DESIGN PROCESS	13	72	5	17
PART OF BOTH THE PROGRAMMING AND DESIGN PROCESSES	4	22	21	70
SEPARATE FROM THE PROGRAMMING AND DESIGN PROCESSES	0	0	0	0
SAMPLE SIZE	18		30	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

Force or major command policy may dictate a certain approach. The question was unaltered for Group A because the researcher assumed these respondents use the approach they believe is best. The results for Question 50 are in Table 78. Only Group A reached consensus on the question with 61 percent choosing the segregated approach as their method. Group B was divided among the possible answers.

The next three questions (26, 27, and 28) are related to the approaches listed in Question 50. Only the Group A questionnaire contained Question 26. This question asked if programming should be completed prior to design (Table 79). In the second round, Group A (90%) strongly supported the

TABLE 78

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 50

The Architect's Guide to Facility Programming lists three basic approaches to programming, which of the following approaches best describes your programming method.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
SEGREGATED	11	61	6	35
INTEGRATED	3	17	2	12
INTERACTIVE	2	11	7	41
SEGREGATED OR INTERACTIVE	2	11	1	6
SAMPLE SIZE	18		17	
CONSENSUS	YES		NO	

TABLE 79

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 26

Programming should be completed prior to design.

	GROUP A		GROUP B*	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	15	75	13	43
(4) AGREE	3	15	13	43
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	2	10	3	10
(1) STRONGLY DISAGREE	0	0	1	3
SAMPLE SIZE	20		30	
MEAN	4.550		4.133	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.944		1.074	
CONSENSUS	YES		YES	

* Data from Round One Questionnaire

statement. Question 27 inquired whether programming should be integrated with conceptual design (Table 80). The original question was changed by replacing "design" with "conceptual design" in the sentence. Even with the change, neither group achieved consensus on the question. Group B, though, did show a bias towards "agreeing" with the statement with 65 percent. Question 28 asked if programming and conceptual design should be interactive (Table 81). Question 28 was also altered by replacing "design" with "conceptual design" in the statement. Group B (90%) "agreed" with the statement. In comparison, Group A did not reach a consensus, though a majority of the respondents (60%) supported the idea that programming and conceptual design should be interactive.

TABLE 80

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 27

Programming should be integrated with conceptual design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	1	5	2	10
(4) AGREE	7	35	11	55
(3) UNDECIDED	1	5	1	5
(2) DISAGREE	10	50	6	30
(1) STRONGLY DISAGREE	1	5	0	0
SAMPLE SIZE	20		20	
MEAN	2.850		3.450	
MEDIAN	2.000		4.000	
STANDARD DEVIATION	1.137		1.050	
CONSENSUS	NO		NO	

TABLE 81

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 28

Programming and conceptual design should be interactive, not separate phases of the facility delivery process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	4	20	3	15
(4) AGREE	8	40	15	75
(3) UNDECIDED	2	10	0	0
(2) DISAGREE	4	20	2	10
(1) STRONGLY DISAGREE	2	10	0	0
SAMPLE SIZE	20		20	
MEAN	3.400		3.950	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	1.314		0.759	
CONSENSUS	NO		YES	

The last three questions in this section, continue to examine programming's relationship with design. Question 31 stated that end product of programming is information, not design (Table 82). Both Group A and Group B supported the statement with 100 percent and 90 percent of the respondents "agreeing", respectively. Question 40 asked whether the programming - design connection can be a problem (Table 83). The original question was altered by replacing the phrase "recurring problem" with "can be a problem." With the change, Group A (89%) supported the statement. In comparison, Group B did not achieve a consensus. However, a majority of Group B respondents (60%) did "agree" that the programming - design relationship can be a problem.

TABLE 82

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 31

The end product of programming is information not design.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	18	90	3	15
(4) AGREE	2	10	15	75
(3) UNDECIDED	0	0	0	0
(2) DISAGREE	0	0	1	5
(1) STRONGLY DISAGREE	0	0	1	5
SAMPLE SIZE	20		20	
MEAN	4.900		3.900	
MEDIAN	5.000		4.000	
STANDARD DEVIATION	0.308		0.912	
CONSENSUS	YES		YES	

TABLE 83

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 40

During the facility delivery process, the programming - design relationship/connection can be a problem.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
(5) STRONGLY AGREE	1	5	1	5
(4) AGREE	16	84	11	55
(3) UNDECIDED	1	5	3	15
(2) DISAGREE	1	5	5	25
(1) STRONGLY DISAGREE	0	0	0	0
SAMPLE SIZE	19		20	
MEAN	3.895		3.400	
MEDIAN	4.000		4.000	
STANDARD DEVIATION	0.567		0.940	
CONSENSUS	YES		NO	

Finally, Question 45 asked if programming was either part of or separate from the design process (Table 83). Neither group achieved consensus on the question. Both Group A and Group B were divided among the possible responses.

TABLE 84

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 45

Programming is ____________ the design process.

	GROUP A		GROUP B	
	FREQ	PERC	FREQ	PERC
PART OF	4	20	9	50
SEPARATE FROM	8	40	9	50
INTERACTIVE WITH	8	40	0	0
SAMPLE SIZE	20		20	
CONSENSUS	NO		NO	

Programming Techniques

In the first round, three questions were asked to determine which programming techniques were most widely used. In the second round, similar questions were asked to ascertain which techniques the respondents thought were most effective. Only techniques that achieved a 50 percent and 40 percent response rate for Groups A and B were listed, respectively.

Question 52 asked which techniques were most effective in collecting programming information (Table 85). The list of possible answers included 9 techniques for Group A and 5

TABLE 85

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 52

Which of the following techniques are most effective when collecting programming information.

	GROUP A		GROUP B	
TECHNIQUE	FREQ	PERC	FREQ	PERC
Interviews	20	100	17	89
Direct Observation	19	95	14	74
Background Data Research	17	85	14	74
Surveys	16	80	6	31
Questionnaires	19	95	NA	
Participant Observation	6	30	5	26
Standardized Data Forms	12	60	NA	
Data Logs	3	15	NA	
Preference Matrix	7	35	NA	
SAMPLE SIZE	20		19	

techniques for Group B. For Group A, 50 percent of more of the respondents thought 6 of the 9 techniques were effective. 5 of these techniques fell into the research and background methods category. Only one fell was an observation method. In comparison, 50 percent or more of the Group B respondents thought 3 of 5 techniques were effective. Of the 3, 2 were research and background methods and 1 was an observation method.

Question 53 requested which techniques were most effective for analyzing and organizing programming data (Table 86). The list of possible responses included 19 techniques for Group A, and 10 techniques for Group B.

TABLE 86

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 53

Which of the following techniques are most effective when analyzing and organizing programming information.

	GROUP A		GROUP B	
TECHNIQUE	FREQ	PERC	FREQ	PERC
Project Cost Estimating	13	65	15	79
Bubble Diagram	17	85	7	37
Construction Cost Estimating	9	45	7	37
Life Cycle Cost Analysis	8	40	10	53
Functional Relationship Diagram	18	90	10	53
Flow Diagram	16	80	2	10
Organizational Chart	15	75	4	21
Space Unit Standards	16	80	7	37
Space Program	16	80	NA	
Layout Diagram	3	15	13	68
Descriptive Statistics	15	75	NA	
Bar Chart/Milestone Chart	10	50	NA	
Relationship Matrices	14	70	NA	
Worksheets	5	25	2	10
Adjacency Diagram	16	80	NA	
Critical Path Method	4	20	NA	
Block Diagram	5	25	NA	
Activity Time Chart	7	35	NA	
Inferential Statistics	5	25	NA	
SAMPLE SIZE	20		19	

50 percent or more of the Group A respondents indicated 11 techniques as most effective. For Group A, 5 were correlation diagrams and 2 were space analysis techniques. The remaining 4 techniques included a cost analysis technique, a scheduling technique, a statistical analysis technique and relationship matrices. In comparison, for

Group B, 50 percent or more of the respondents indicated 4 techniques as most effective. Of the 4, 2 were cost analysis techniques and 2 were correlation diagrams.

Question 54 asked which techniques the respondents thought were most effective for communicating and evaluating programming information (Table 87). The possible answers were a list of 10 techniques for Group A, and 8 techniques for Group B. For Group A, 50 percent or more of the respondents showed a preference for 4 of the 10 techniques. All 4 were presentation and documentation methods. For Group B, 50 percent or more of the respondents indicated 4

TABLE 87

ROUND TWO DESCRIPTIVE STATISTICS FOR QUESTION 54

Which of the following techniques are most effective when communicating and evaluating programming information.

	GROUP A		GROUP B	
TECHNIQUE	FREQ	PERC	FREQ	PERC
Graphics	20	100	12	63
Oral Presentations	18	90	11	58
Narrative	13	65	12	63
Brainstorming	9	45	14	74
Group Planning	9	45	7	37
Audio/Visual Aids	13	86	1	5
Panel Discussions	NA		3	16
Forums	3	15	NA	
Evaluation Matrix	5	25	NA	
Buzz/Rap Session	NA		1	5
Weighting	5	25	NA	
Rating and Rating Scales	5	25	NA	
SAMPLE SIZE	20		19	

of 8 techniques as most effective. Of these 4, 3 were presentation and documentation methods and 1 was a collective decision making method.

Chapter Summary

Chapter V summarized the results of the Round Two Delphi Questionnaire. The data included the descriptive statistics on each second round question including: (1) response frequencies, (2) response percentages, (3) the mean, (4) the median, (5) the standard deviation, and (6) sample size. The second round concluded the data gathering portion of the research.

The next chapter analyzes the data collected from both rounds of questionnaires. The analysis includes comparing the two research groups, and drawing conclusions about their similarities and differences. The researcher uses the information accumulated from the questionnaires' statistical data, the written comments and answers, and the literature review to determine strengths and weaknesses in the Air Force programming procedures.

VI. Analysis

Chapter Overview

This chapter analyzes the several aspects of facility programming. The researcher uses the information collected from the literature review and the Delphi questionnaires, including the statistical data and written responses. First, a comparison of the two research groups is included to point any significant differences in attitudes. The remaining sections examine programming in 4 areas: (1) programming's purpose and information requirements, (2) the roles of programming participants, (3) the interaction between programming and design, and (4) programming techniques.

Comparison of Groups A and B

Two of the research objectives were to identify the weaknesses and strengths of the programming processes used by the Air Force and commercial firms. One way to accomplish these goals were to compare the attitudes and beliefs about programming from "expert panels" representing the two groups. The groups's differences and similarities were measured using two approaches. First, the demographic questions asked in round one of the survey instruments examined educational backgrounds and respondent experience. Second, Wilcoxon Rank Sum tests were performed on the questions using the five-point Likert scale. The Likert scale questions measured the groups' views on different aspects of programming.

The demographic questions revealed two notable differences between Group A and Group B in the following areas: (1) educational backgrounds, and (2) programming experience. The group composed of professionals outside the Air Force all had backgrounds in architecture and significantly more experience in programming. An overwhelming majority of the Air Force group had backgrounds in engineering, especially civil engineering. Also, their experience was spread out over the areas of programming, design, and construction.

Table 88 shows the results of the Rank Sum test on the 35 Likert scale questions. The tests measured whether there were significant differences in the groups' means at a significance level of 0.05. The results show that Group A and Group B had different attitudes on 18 questions. The researcher examined the contents of each question and grouped them into categories. The two respondent groups showed few significant differences on questions dealing with (1) the roles of programming participants, and (2) programming approaches or methods. The differences appeared in questions about the role of the programming document and the types of requirements programming identifies.

However, the Rank Sum results must not be misinterpreted. The tests do not reveal if the groups

TABLE 88

COMPARISON OF GROUPS A AND B
USING RANK SUM TEST

QUESTION	P VALUE	DIFFERENCES
6	0.0018	YES
7	0.0003	YES
8	0.0000	YES
9	0.0010	YES
10	0.0102	YES
11	0.6572	NO
12	0.0015	YES
13	0.0082	YES
14	0.9676	NO
15	0.1906	NO
16	0.7301	NO
17	0.1846	NO
18	0.3104	NO
19	0.0089	YES
20	0.0000	YES
21	0.8799	NO
22	0.9784	NO
23	0.0337	YES
24	0.2096	NO
25	0.0010	YES
26	0.0797	NO
27	0.1264	NO
28	0.2616	NO
29	0.3793	NO
30	0.0006	YES
31	0.0000	YES
32	0.0038	YES
33	0.0001	YES
34	0.0135	YES
35	0.0179	YES
36	0.0491	YES
37	0.5434	NO
38	0.6324	NO
39	0.6260	NO
40	0.1292	NO

"agreed" or "disagreed" on a question, and they do not show if the groups met the consensus criteria. For example, on Question 6, both groups reached consensus and "agreed" with the statement. However, the groups' means were significantly different accounting for the degree of conviction. The reason is the use of a five point Likert scale that allowed the respondents to "agree" or "strongly agree" on the question. In another example, on Question 27, Group A did not achieve consensus, but Group B did reach consensus. Still, the Rank Sum test showed no significant difference between the groups.

The Purpose of Programming

Clearly, the purpose of facility programming is the identification of the necessary requirements for completion of a project. The question is what types of requirements and how much information should programming identify. In addition, what method or vehicle is used for transmitting the programming information.

Outside the Air Force, programming professionals usually produce a formal document containing the project requirements. However, Air Force personnel do not always generate a programming document for each project. One possible reason is that the Air Force does a large number of simple, low cost projects in-house where programming is done in an informal fashion. In contrast, an outside programming or design firm is hired to provide services on larger, often more complex facilities.

Both respondent groups agree that programming includes the major issues to be addressed in the conceptual design phase, though not necessarily the details for design development. The major issues can involve (1) organizational goals and objectives, (2) functional requirements, (3) technical requirements, (4) budget and cost information, (5) environmental concerns, (6) energy goals and objectives, and (7) scheduling concerns. However, the two groups appear to emphasize different requirements. When examining the research data, clearly both outside and within the Air Force programming identifies the necessary functional information. Differences did appeared, though, when the groups were asked what information a programming document should include. The Air Force participants unanimously answered budget and cost information. In contrast, organizational goals and functional requirements were the most frequent responses by the outside programming "experts." In comparison, the same group ranked cost requirements a distant third, while organizational goals were last on the Air Force "experts" list. In addition, for the Air Force respondents, environmental data and energy requirements were a strong third and fourth on their list, respectively. This supports the Air Forces long standing concern, since the late seventies, with facility energy consumption and their relatively recent push for remediating and preventing environmental hazards.

Four of the questions in the survey dealt specifically with (1) functional, (2) technical, (3) quantitative, and (4) qualitative requirements. As previously mentioned, both groups support the fact that programming identifies functional requirements. However, only the respondents outside the Air Force thought technical requirements should be included. The comments from both groups, though, revealed that in many cases only special or unique technical requirements should be identified. In addition, the level of detailed technical information is most often left to the designer. Both groups, also, supported including quantitative and qualitative requirements. However, by examining the frequency of responses, the Air Force respondents seemed to emphasize quantitative information, in lieu of the qualitative requirements.

Another important aspect when exploring programming's *function*, is who uses this information. Two questions asked if the programming document was primarily information for the designer or for the client/user, respectively. Neither group supported the notion that the information was "primarily" for either individual or group. However, through their written comments, over half of the programming "experts" expressed the idea that the information was important to both the client and the designer. For the client, the programming information aids in project decision making. While, for the designer, the information sets the direction for design and often confirms program concepts. On the other hand, the researcher could not draw the same conclusions from the Air Force respondents. Their written comments, though, did identify another primary recipient of programming information, approval authorities. This group is unique to the Air Force and other DOD agencies. Most large projects need funding and approval from some source, usually a major command, Air Staff, or the U. S. Congress.

The researcher concludes that the purpose of programming is primarily to identify functional requirements, both quantitative and qualitative. In addition, the scope or nature of a construction project may dictate including other types of information. However, many of the Air Force respondents see the programming process mainly as producing a funding and approval document, notably a DD Form 1391. In fact, in their written comments, 10 of the 31 Chief Engineers refer to programming in terms of justification and budgeting. This, in part, explains the high frequency of responses supporting quantitative, and cost information in the programming document.

Programming Participants

The individuals or groups directly involved with the programming process fall into 4 main categories: (1) programmers, (2) clients or "customers", (3) users, and (4) designers. In addition, these "roles" often overlap with clients and designers performing the role of programmers. In another example, clients and users are frequently the same group. In fact, in the Air Force, it is conceivable the Civil Engineering organization could include people filling all four functions. Another significant player in the programming process is the organization providing funding and approval authority for a project. From private practice's point of view, this is usually the client or owner. However, within the Air Force structure, the agency with approval authority often is a command headquarters or higher. The important difference is that this organization frequently is not an active contributor in programming process.

Clients and Users. A discussion of the client versus the user is necessary to understanding programming roles. Clearly, client and user participation is essential to any successful programming effort. They both are important contributors of programming data that includes individual preferences, behaviors, and perceptions, as well as organizational activities, structure, and objectives. However, one notable difference is that the individual or group acting as the client, normally, has the decision making power over the project. In comparison, the users, which may or may not include the client, include everybody performing a function in that facility. The primary users are the building occupants, but other users could include the occupants' "customers", maintenance or janitorial personnel, or the general public. The drawback is that the users, potential sources of valuable information, often are ignored. In the Air Force, the differences between users and clients are more notable. First, the term "client" is not widely used. The "using agency" is Civil Engineering's "client." The word "customer" is frequently substituted. Second, decision making power over a facility project is spread out over the using agencies' commander, Civil Engineering, and the appropriate approval authority. Third, the military rank structure is a powerful influence in setting project requirements. In the Air Force respondents' comments, one complaint was the deference to an individual commander's preferences, though it may not represent the best solution to particular problem. It is easy to see how the individual user's, maybe a secretary's or an airman's, legitimate ideas or concerns could be overlooked.

Responsibility and Control. Another important issue is who is responsible for facility programming and who controls the programming process. Traditionally, from the architecture community's point of view, the client or owner is responsible for the "program." Group A, the programming "experts", supported this notion. The Air Force respondents' did not agree, at least in one respect. For Civil Engineering, the using agency is their "client." As the using agency's representative in facility acquisition, Civil Engineering's in-house programming staff has the responsibility and control over programming. From another perspective, the Air Force research participants, endorse private practice's view. Often facility projects are designed by commercial Architecture-Engineering firms. In these cases, Civil Engineering acts as their client and produces the initial "program." For the professional programmers outside the Air Force, the issue involving the "control" of programming was unresolved. The group did agree that a design firm's in-house programming staff should control the process, but only if the client has no in-house programming staff and the design firm has programming expertise. From the written comments, "control" depends primarily on both the client's and design firm's capabilities. Another option, though not supported by either group, is an outside consultant specializing in programming.

Closely linked to the issues of control and responsibility is decision making. Plainly, from the structured and unstructured responses by both groups, the client's have decision making power over many aspects of a facility project. However, this does not carry over to specific or technical decisions on the facility's design. Clearly, design is the creative solution to the problems identified in the programming process. The design decisions, the "nuts and bolts," are the domain of the designer. The clients, though, influence the design process with their inputs during programming. The design must meet their requirements and, more importantly, their approval. During programming, a programmer's function is to guide a client through the decision making process that sets the requirements for design. Though a client's decisions, during programming or design, are extremely important, getting clients to make decisions is a recurring problem. Another point of view, brought out in the respondents' comments, is that this is a "challenge" more than a "problem."

<u>Team Concept</u>. Clearly, programming is a "team" effort involving the programmer, designer, client and users. All four have valuable skills or knowledge to contribute to the programming process. First, the programmer has expertise in the techniques of collecting, analyzing, organizing, evaluating, and presenting data. Second, designers, as recipients of the "program", are important contributors of ideas and information. Third, the client and users are often the primary source of programming data, since they are "experts" on the organization's functions and activities. A successful programming effort requires the active support and participation of all four individuals or groups.

At this point, a discussion of the programmer versus the designer is valuable in understanding their roles as members of a programming team. From the research, designers are not responsible for nor do they control the programming process. This is the domain of the programmers, whether they are part of the client's, design firm's or outside consultant's organization. Further, if programming is thought of as "analysis" and design as "synthesis," it is important to realize the two functions require different skills and thought processes. However, the programmer, or someone on the programming team, should have experience in design and understand the facility delivery process. One compelling reason, from the literature review, is that an effective program must include relevant information, and present that information in a usable format, for the designer. In part this explains why programming is a growing architectural service. Architects are trained as programmers because they often are knowledgeable of the multi-discipline design requirements of architects and engineers. In addition, they have a broad understanding of the facility acquisition process.

Communication between the programmer, designer, client, and users is essential for effective programming. The programmer, directing the programming process, should have well developed communication skills, including graphic analysis and presentation. To aid in the communication process, programmers should educate the client and users in the programming process. In addition, educating the client/user in the architectural design process is also desirable. An understanding and appreciation of programming and design facilitates effective communication of meaningful information and the necessary support for the programming effort. However, depending on the client/user's experience, the required level of education will vary.

Programming and Design

A majority of the research concentrated on programming and design, and their interaction with each other. From the literature review, one of the biggest controversies involves the relationship between these two processes. However, the research did reveal that both are parts of a problem solving cycle. Programming defines or states the problem, while the design represents the solution. Though this is a simple concept, it aids in understanding or clarifying the different roles of programming and design.

Programming and design are both part of the facility delivery process. Delivery, in the broad sense, meaning the completed renovation or new construction of a facility, usually a building, for some stated purpose. The research groups agreed that distinct phases of this process are: (1) programming, (2) conceptual design, (3) design development and contract documents preparation, (4) construction.

Further, programming is an essential part of the facility delivery process, even though it may only comprise 5 to 15 percent of the overall project development time. In addition, some programming is done on all projects. However, the outcome is not always a formal programming document.

In addition, programming and design are both processes within the facility delivery process. However, where the programming process ends and the design process begins is not always clear. One similarity between programming and design is that they are both iterative processes. Iterative meaning two or more cycles of information review, evaluation, and feedback. For the Air Force respondents, formal program and design reviews occur approximately 3 and 2 times with the user, respectively. In comparison, the outside programming "experts" indicated an average of 4 program reviews and 3 design reviews with their clients. The difference in the number of reviews between the groups, though, is unclear. Possible reasons for the lower Air Force statistics include (1) funding constraints, (2) project types, (3) project complexity, (4) project size, and (5) unstructured reviews. The last reason requires some explanation, and points out a potential problem. Air Force programmers and the users are usually stationed at the same base facilitating unplanned and, often, unrecorded dialogues between the two groups. However, the commercial programming or design firm normally is not located in close proximity to their clients. When they meet with their clients, the efficient and effective use of time is important to the firm's success. This means preparation and planning, as well as accurate notes of the proceedings. In other words, though the Air Force programmer may actually talk with the user quite often, important information may be overlooked or unrecorded because of the informal nature of the meetings.

Another point on which the two groups agreed, was that the programming process is not the same for all projects.

The information requirements for each new construction project are unique. This often means taking a different approach or using diverse techniques to gather, analyze, organize, evaluate and present programming data. For the Air Force, the different programming criterion are apparent in their numerous facility programs that include: (1) Operation and Maintenance (O&M), (2) Military Construction (MILCON), (3) Non-Appropriated Fund (NAF), and (4) Military Housing (MFH).

Further, the interaction between programming and design is important because the designer must comprehend and respond to the programming data appropriately. A smooth transition is essential to insure the relevant project requirements are not ignored or lost. However, the program should not necessarily dictate solutions nor inhibit the designer's creativity in producing the design. As mentioned previously, there are three basic approaches to programming - design relationship: (1) segregated, (2) integrated, and (3) interactive. A clear majority of the Group A respondents, or programming professionals, use a segregated approach. In comparison, the Air Force "experts" thought a segregated or interactive approach was best. Also, both respondent groups supported the view that programming should be completed prior to initiating design, at least in an ideal situation. Design, in this context, meant design development and contract documents production. However, the particular approach used for a project does depend on its requirements. The best example is when schedule constraints do not allow time for separate programming and design efforts, encouraging an interactive approach.

The researcher draws one strong conclusion about the programming - design connection, that conceptual design is the link between the two processes. Conceptual design, for the purposes of this research, was equated to schematic design. However, within the design profession, conceptual design and schematic design have different meanings. For some, the terms are interchangeable, for others, they are two different exercises, usually involving the amount of design detail. Nevertheless, conceptual and schematic designs both explore functional design solutions and precede design development.

The question, then, becomes how does programming interact with conceptual design. The two groups did not agree on how conceptual design fits into the programming and design processes. For the programming "experts" outside the Air Force, conceptual design is not part of programming, but is part of the design process. However, a majority of Group A respondents acknowledge the usefulness of an interactive relationship between programming and conceptual design. One respondent wrote: "They [conceptual design and programming] can be mutually supportive and time saving to do coordination with schematics (Appendix E)." Others see conceptual design as a way to test the validity of the programming information prior to design development. The Air Force participants view conceptual design as part of both the programming and design processes. In addition, programming and conceptual design are interactive. One probable reason, for the Air Force group's responses, is that a conceptual, single-line drawing of a possible facility solution, in the past, has been required in the DD Form 1391, the funding approval document. In fact, one respondent wrote: "The programmer must have a good idea of the probable solution to have his cost estimate within 25 percent of the final CWE [Construction Working Estimate] (Appendix F)." Again, this points out Air Force's emphasis on the cost estimating and approval aspect of programming. However, regardless of whether conceptual design is part of the programming process, the end product of programming is information, not design.

In closing the discussion on interaction between programming and design, the researcher stresses the importance of this relationship. The transmitting of pertinent project information to the designer is critical. Responding to the statement that "the programming - design connection can be a problem", a majority in both groups agreed. In fact, the Group B respondents indicated that the amount of time elapsed between programming and design, using Air Force programming methods, was a problem in adequately defining project requirements.

Programming Techniques

Programming techniques are the ways programmers collect, analyze, organize, evaluate and communicate information. Often these techniques are methods or processes in themselves. A number of different techniques are usually used during programming. The research attempted to discover which techniques were widely used, and subsequently which were most effective. The survey instruments listed 70 techniques in three main areas: (1) information collection, (2) information analysis and organization, and (3) information evaluation and communication. A 50 percent or more response rate was the criteria used to determine if a technique was widely used, or considered most effective by the respondents.

Overall, the research revealed two notable differences between the two participant groups. First, the Air Force respondents had lower response rates in most areas. Second, the Air Force group had fewer techniques meet the research criteria as widely used or effective. Possible reasons for the differences include: (1) a lack of familiarity with many of the techniques, and (2) the Air Force's emphasis on the cost and approval aspects of programming. Another underlying reason could be the different educational backgrounds of the respondent groups. The professional programmers outside the Air Force all have architectural training, while the Air Force group were almost entirely engineers by trade. Architects normally have some exposure to programming methods or techniques, in school or on the job. In comparison, the educational process for engineers is somewhat different, and usually does not include programming.

Data Collection. Data collection techniques fall into three categories: (1) research or background methods, (2) observation methods, and (3) comparison methods. For both groups, the most widely used techniques were the research and observation methods. In addition, the most effective methods are listed below in order of response frequency. For Groups A, the most effective techniques were:

1. Interviews
2. Questionnaires
3. Direct Observation
4. Background Data Research
5. Surveys
6. Standardized Data Forms

For Group B, the techniques were:

1. Interviews
2. Background Data Research
3. Direct Observation

The familiarity with and perceived effectiveness of the research and background techniques is not surprising. They are considered the primary means of collecting data from clients and users, and any programming effort includes at least one of these techniques.

Analysis and Organization. The largest number of different techniques are used for analyzing and organizing programming data. They fall under a number of subcategories, as follows: (1) statistical analysis, (2) functional and activity analysis, (3) space analysis, (4) cost analysis, (5) scheduling, (6) relationship matrices, and (7) correlation diagrams. For the outside programming "experts", the list of widely used techniques fell into 6 of the 7 subcategories, the majority being correlation diagrams and space analysis techniques. In comparison for the Air Force participants, their list included only cost analysis techniques and correlation diagrams. In addition, the most effective methods, using the research criteria, are listed below by group and response frequency. For Group A, the list included:

1. Functional Relationship Diagram
2. Bubble Diagram
3. Space Program
4. Space Unit Standards
5. Flow Diagram
6. Adjacency Diagram
7. Descriptive Statistics
8. Organizational Charts
9. Relationship Matrices
10. Project Cost Estimating
11. Bar/Milestone Charts

For Group B, the Air Force participants, the most effective techniques were:

1. Project Cost Estimating
2. Layout Diagram
3. Functional Relationship Diagram
4. Life-Cycle Cost Analysis

The research data on analysis techniques appears to support the hypothesis that the Air Force emphasizes the cost and approval aspects of programming. In comparison, the group of participants outside the Air Force stress the organization's functional and space requirements during programming. One of the most significant differences between the two groups was concerning the space analysis techniques. For the "experts" outside the Air Force, the space program and space unit standards were the numbe. one and two most used techniques. In comparison, neither of these techniques were used by more than 41 percent of the Air Force respondents. However, in the literature, space was described as "the single most important element of a facility" and all other programming elements depend on the physical characteristics of space (A:99).

Evaluation and Communication. The evaluation and communication techniques fall into three subcategories: (1) collective decision making techniques, (2) presentation and documentation techniques, and (3) rating techniques. For the respondent group outside the Air Force, the most widely used techniques fell into all three subcategories. For the

Air Force respondents, the collective decision making, and presentation and documentation techniques were the most used. The list of the most effective techniques was similar for both groups. For the programming "experts" outside the Air Force, the techniques were:

1. Graphics
2. Oral Presentations
3. Audio-Visual Aids
4. Narratives

For the Air Force respondents, the list in order of response frequency was:

1. Brainstorming
2. Graphics
3. Narratives
4. Oral Presentations

The above lists only includes presentation and documentation techniques, except for brainstorming which is a collective decision technique. For the outside programming "experts", graphics was unanimously included as an effective technique. This seems to support their emphasis on correlation diagrams. Most correlation diagrams are graphical ways to analyze, organize and communicate programming information.

In closing the discussion on programming techniques, if a particular technique did not meet the criteria as widely used or effective does not necessarily mean that the technique is not useful or effective. Many of the techniques have very specific uses, methods and results. For example, inferential statistics is a complex mathematical technique. Most programming efforts would never require its use. However, on a large, diverse project it may be valuable. In another example, not all programming efforts include a schedule analysis, but if time is a critical factor one of the scheduling techniques may be useful. In comparison, most programming endeavors do involve some research, functional analysis, cost analysis, presentation and documentation, explaining, in part, which techniques met the research criteria for use and effectiveness.

Chapter Summary

Using the research data, similarities and differences between the two research groups, as well as common elements in programming were established. First, the programming process identifies functional requirements. However, the Air Force programming methods, also, emphasis preliminary cost estimating to support funding approval. Second, programming is a team effort involving the programmer, designer, client, and user. In addition, communication and education are important using the team concept. Third, the programming - design relationship is critical. Though, programming approaches may differ, the link between the two is conceptual design. Finally, the research established a list of widely used, and effective programming techniques.

The next chapter builds upon the research analysis. It identifies potential problem areas in Air Force programming. In addition, the researcher recommends ways to improve the programming process.

VII. Conclusions and Recommendations

Chapter Overview

In the final chapter, the researcher uses the research .ata to draw conclusions about the Air Force programming processes. The conclusions discuss Air Force programming methods and point out possible areas of improvement. The researcher follows by making recommendations for improvements in two areas: (1) the programming process, and (2) education and training. The researcher's primary proposal is a new programming model. The researcher concludes with suggestions on the testing of the model.

Conclusions

The researcher started research into the area of facility programming for two reasons: (1) the researcher's own perceptions of inadequacies in the current Air Force process, and (2) the researcher's desire to find better ways to produce quality facilities meeting the user's needs. Also, programming seemed a logical place to start improving the Air Force's design and construction process. First, effective decision making at the project's inception has a positive impact on the design and construction phases. This translates into better "customer" satisfaction, fewer design and construction changes, and reduced costs. Second, if the researcher had examined the design or the construction phases, the problems discovered may have actually been symptoms of poor identification of project requirements.

The researcher uncovered several areas where improvements could be made to the programming process. One area is in the identification of the using organization's functional requirements for design. The Air Force emphasizes the project funding and approval aspects of programming. The functional requirements identification, though stated as one of programming's objectives, is secondary.

Further, the Air Force produces primarily two kinds of programming documents: (1) Military Construction Project Data (DD Form 1391), and (2) Project Book or Project Definition. The DD Form 1391 is a relatively short document, one to three pages, including a preliminary cost estimate, project requirements, and a description of the proposed construction. Though called a programming document it's primary use is to request and justify a construction project. The form leaves little room to identify functional requirements in any detail. In fact one respondent wrote: "The major problem I see is the level of information needed for design cannot be included on the 1391 (Appendix F)." However, the DD Form 1391 is the main programming document for most Air Force construction projects.

The Project Book, on the other hand, is usually a lengthy document containing the "design data, criteria, major command policies, functional requirements and cost information ... for facility projects (1:29)." In contrast, Project Books are very structured, in a checklist type format. They, also, contain detailed technical information. Neither of these characteristics, though, are desirable according to the research data. First, most construction projects are unique, and "programs" should be individually formatted to include only the relevant information for that project. Second, the programming document should only contain special technical requirements. The detailed design information is left to the designer. However, the Project Book is being replaced by the new document called a Project Definition. The Project Definition is part of the new Requirements and Management Plan (RAMP) created to improve MILCON execution. The Project Definition seems to respond to concerns identified in the research, however, the RAMP concept is brand new and still untested.

The Project Book or Project Definition, without a doubt, does a more thorough, better job, than a DD Form 1391, in identifying functional requirements. However, they are only required for MILCON projects. Though MILCON represents the largest portion of the Air Force's construction dollars, often the Operation and Maintenance (O&M) construction program represents the greater workload for Base Level Civil Engineering organizations. In other words, for O&M projects, the documentation of functional requirements is primarily accomplished on the DD Form 1391.

The research, though, indicates the effective functional programming is essential to a project's success regardless of project size.

In addition, the way the Air Force programs and designs renovation or new construction projects often encourages exploration of only a single solution, though other, perhaps better options are available. First, the DD Form 1391, normally includes a single, line drawing of a design solution. Though, the drawing may or may not represent the final design, it sets a strong precedence when entering the design development stage, thus discouraging other solutions. These drawings, though still required by regulation, are now highly discouraged. Second, this potential drawback is more pronounced when the design services are performed by commercial Architect-Engineering (A-E) firms. A-E firms are normally hired under a negotiated, firm fixed-price contract. Since the fees are fixed, the design firm is discouraged from proposing more than one or two design solutions. In other words, the more time spent in design development eats into the firm's profit. Further, Air Force design contracts usually do not specifically recognize conceptual or schematic design. The research, though, pointed out that conceptual design is a separate, distinct phase in the facility delivery process, and the important link between programming and design development (Figure 15). In addition, the terms of the contract normally require design submittals at the approximately 30, 60 and 90 percent

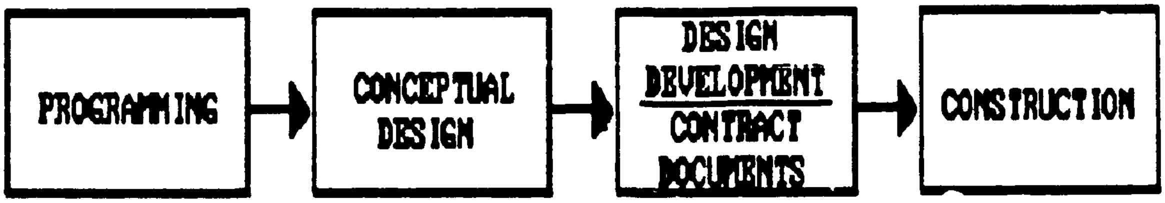

Figure 15. Four Distinct Phases of the Facility Delivery Process

design. However, 30 percent design submittals often include more than just floor plans and elevations. Changes, at this point, to the functional layout or aesthetics of a facility can involve some redesign in other areas. The new RAMP addresses some of these issues. The new Project Definition requires a maximum of three architectural concepts with corresponding floor plans and elevations. Again, however, the O&M program is not included.

Another characteristic of programming and design, from the research data, is that they are iterative processes. Iterative meaning one or more cycles of review, evaluation and feedback either on the programming information or design schematics. The literature in the Air Force programming and design process does not appear to address the iterative process. Further, it is unclear whether the new RAMP process accounts for more than one cycle of review, evaluation and feedback. Iterations are necessary to refine programming information or the basic design solution, and should be recognized.

Ways of shortening the time between programming and design is another area for improvement. The amount of time between the two processes was one of the most frequent responses to inadequacies in identifying project requirements. DD Form 1391s and, in the past, Project Books were prepared months ahead of design initiation. The time "lag" required reexamination of project requirements at the design stage. Another problem area, related to the time interval, were user changes. Air Force personnel are moving all the time. New users, during the programming and design process, bring their own personal preferences and attitudes. Depending on the person, this could mean changes in programming requirements or conceptual design. A lengthy interval between programming and design almost assured manpower changes within the using agency.

Another problem, identified by the Air Force research respondents, was programmers' lack of experience. Often new officers or lower-grade civilians fill the programmer position. This is also aggravated by the research findings showing a lack of familiarity with a wide range of programming techniques. Education and training are connected to this problem. First, the majority of the higher-grade civilians and officers in Civil Engineering are engineers, by education or experience. Programming methods or techniques are not normally taught to engineering students, nor is programming a service usually provided by engineers. Second, architectural programming techniques are only covered briefly in continuing education courses provided by the School of Civil Engineering and Services at the Air Force Institute of Technology.

The last point under conclusions, is the importance of team work and communication. The success of any programming effort lies in these two areas. First, programming should be accomplished by a programming team, composed of the programmer, designer, and client/user. Second, communication facilitates this team concept. The programmer should have well developed communication skills. An ability to accurately portray data, both orally and visually, will prevent misunderstandings. He should, also, educate the user in the programming process. A user who understands and appreciates the information requirements can better communicate them. Finally, the designer is a primary recipient of programming information. As such, the designer can state what information and format is most useful to him.

Recommendations

The following are the researcher's recommendations for improving Air Force programming processes. The researcher bases his ideas on the data gathered from the research effort. The recommendations fall under three areas: (1) the programming process, (2) education and training, and (3) testing

The Programming Process. The researcher proposes a generic programming model applicable to the MILCON, O&M, and other Air Force construction programs. The programming model is a combination of the segregated and interactive approaches described throughout the thesis. The programming process would include two distinct phases: (1) Project and Funding Approval, and (2) Functional Programming (Figure 16). The project and funding approval stage would include the Military Project Data (DD Form 1391), leaving this process intact. The researcher saw no reason to change the DD Form 1391, because it is an established document that accomplishes the mission of gaining project funding and approval. However, the model adds a second programming stage, functional programming. *Functional programming* would include the in-depth examination of the using organization's goals, objectives, and functional requirements. Project budget information can be included, but the document should not become a cost estimating exercise. Other types of

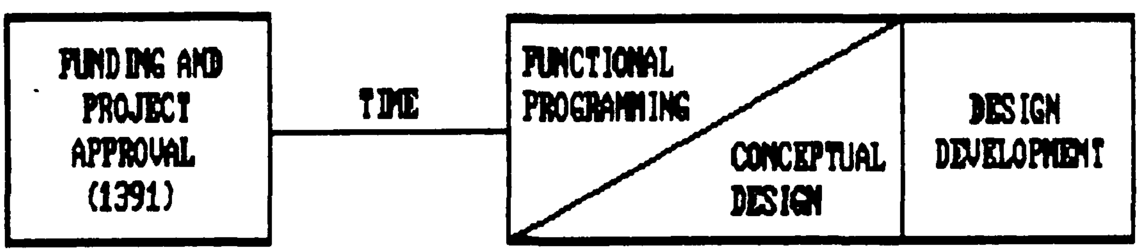

Figure 16. Proposed Programming Model

information contained in the "program" would entirely depend on the project's goals. This might include: (1) technical information, (2) schedule information, (3) environmental data, or (4) energy requirements. There are several important aspects to functional programming phase. First, the programming effort would occur interactively with Conceptual Design. In other words, programming and conceptual design would develop in alternating sequences to each other. Second, functional programming would begin just prior to design initiation. Third, the end product would be a programming document. The format of the document would be determined by the project requireme ts. The document need not be lengthy. A simple project may only require a couple of type written pages. However, use of graphics is recommended since they are quick and effective ways of transmitting information. Fourth, programming would be accomplished by a team including the programmer, designer and users. The team members would be set before starting the process. Finally, the Functional Programming stage would end with the using agency approving the programming requirements and conceptual design. The researcher recommends the final conceptual design include a set floor plan and accompanying elevations.

The researcher recommends using the same Architect-Engineering firm for the functional programming when hiring a commercial firm to perform design services. There are several reasons. First, architects are more familiar with functional programming techniques. Second, often a firm's programming staff understands their design staff's information needs. This helps the programmers format the information in a manner useful to the designer. Third, this avoids the added work of selecting two separate firms, one for programming and one for design. The researcher, however, suggests two different contracts. The first would be for functional programming and conceptual design with a follow on contract for design development and contract documents production. The researcher, also, suggests the firm be reimbursed by a cost-plus-fixed fee, or a time and materials contract for functional programming, and a firm fixed-price contract for design development. The benefits of this arrangement allow for the iterative nature of programming and conceptual design. The firm would not be restricted, by cost, from fully developing the programming information and conceptual design. Though, this type of contract may prove more expensive, the trade-off or savings come from fewer changes during design development and construction. Also, another purpose of programming is determining if project is truly needed or can be accomplished within funding limitations. If the answer to either question is no, the two contract system can save time and money by eliminating the follow-on design work. The Air Force may, also, realize savings from a more accurate estimate of design costs for the firm fixed-price, follow-on contract. Another possible benefit would be reduced design reviews during design development, speeding up the process. For example, instead of three reviews at 30, 60 and 90 percent, only two reviews would be necessary at perhaps 50 and 90 percent. In addition, the reviews cou'd concentrate mostly on technical requirements, since the functional requirements were approved in programming stage.

Many of the programming model's benefits described when using an A-E firm are also applicable to in-house design efforts. However, the primary benefits are reduced users changes during design development and construction. The reasons include:

1. Eliminating the time interval between programming and design, reducing the number of possible personnel changes in the using agency.

2. Involving the users as part of the programming team, increasing their interest and participation in the programming effort.

3. Not restricting the number of programming and conceptual design iterations, allowing full development of the programming information and conceptual design.

4. Requiring user approval of the programming information and conceptual design, committing the user to a set course of action.

In addition, fewer user changes should translate into time and money savings by reducing redesign work and construction change orders. Hopefully, the added concentration on user requirement's in the new programming model will, also, increase customer satisfaction by providing a quality facility.

Education and Training. The research pointed to a lack of familiarity with different programming methods and techniques. The researcher has one suggestion for improving the education and training of Air Force programmers. The School of Civil Engineering and Services at the Air Force Institute of Technology should either expand the existing project programming class or add a new facility programming class to the curriculum. The researcher recommends the second course of action, because it would place the needed emphasis on functional programming, not available today. The new or expanded class should emphasize learning the following programming techniques for the collecting, analyzing, organizing, evaluation and documentation of information.

1. Information Gathering Techniques

    (1) Interviews

    (2) Questionnaires

    (3) Background Data Research

    (4) Direct Observation

    (5) Surveys

    (6) Standardized Data Forms

2. Analysis and Organizational Techniques

   (1) Functional Relationship Diagrams

   (2) Bubble Diagrams

   (3) Space Programs

   (4) Space Unit Standards

   (5) Flow Diagrams

   (6) Adjacency Diagrams

   (7) Descriptive Statistics

   (8) Organizational Charts

   (9) Relationship Matrices

3. Evaluation and Communication Techniques

   (1) Graphics

   (2) Oral Presentations

   (3) Narratives

   (4) Audio-Visual Aids

The above lists were composed of the most widely used and effective techniques, as determined from the research. Other categories of techniques that might be included are:

1. Cost Analysis Techniques
2. Scheduling Techniques
3. Collective Decision Making Techniques
4. Rating Techniques
5. Comparison Techniques

Testing. In closing, the researcher recommends testing the model before full implementation of his ideas. Though, the recommended programming model is based on the research data, it's proposed benefits are still only theoretical.

Testing of the model is necessary to determine if the new programming process is actually beneficial to the Air Force. The researcher suggests trying the new process on one or two projects at an Air Force base in the CONUS. If the model is successful, an expanded test should be accomplished before considering full implementation.

Chapter Summary

The conclusions and recommendations conclude the research effort. The researcher studied the facility programming processes used by the Air Force and commercial firms. The research data was gathered in two ways: (1) a literature review, and (2) a Delphi questionnaire technique. The study included two participant groups: (1) Chief Engineers within Base Level Civil Engineering organizations, and (2) programming "experts" working outside the Air Force. Two rounds of questionnaires were administered to the respondents. The survey instruments measured the attitudes and beliefs of the participants on programming issues. The two groups responses were compared, and hypotheses were drawn about their differences and similarities. The research analysis summarized the research effort using information from the literature review and questionnaires. The study ends, with the researcher making conclusions and recommendations regarding improvements to the Air Force's programming methods.

Appendix A: List of Group A Respondents

The following is a partial list of the Group A respondents, the panel of professional programmers outside the Air Force. The list includes 21 of the 24 individuals that participated in the thesis research. The names of the participants and associated information is printed with their permission. A copy of the researcher's letter and the release form requesting permission are attached.

1. DAVID R. BEARD, AIA

   RTKL Associates, Inc.
   Baltimore, Maryland

2. ROBERT BRANDT, AIA

   Haines Lundberg Waehler (HLW)
   Architects, Engineers, Interior Designers, Planners
   New York, New York

3. MICHAEL BRILL

   President
   BOSTI, the Buffalo Organization for Social and Technological Innovation Inc.
   Buffalo, New York

   Professor
   School of Architecture
   State University of New York at Buffalo

4. DAVID CHASSIN, AIA

   Hellmuth, Obataz, Kassabaum (HOK)
   St. Louis, Missouri

5. ROBERTA M. FELDMAN, PhD

   Phd in Environmental Psychology
   Masters of Architecture

   Feldman Consultants
   Chicago, Illinois

   School of Architecture
   University of Illinois at Chicago

6. JAY FARBSTEIN, PhD, AIA

President
Jay Farbstein and Associates, Inc.
San Luis Obispo, California

7. W. JEFF FLOYD, AIA

Sizemore Floyd Architects
Atlanta, Georgia

8. BRYANT P. GOULD, AIA

Bryant Putman Gould, AIA, PC
Englishtown, New Jersey

9. KENNETH M. HARTH, AIA

Kaplan/McLaughlin/Diaz
San Francisco, California

10. DAVID S. HAVILAND

School of Architecture
Rensselaer Polytechnic Institute
Troy, New York

11. JOSEPH HENSLEY, AIA

Brook * Henslsy * Creager Architects
Spokane, Washington

12. CHARLES KURT, AIA

The Durrant Group, Inc.
Dubuque, Iowa

13. RALPH H. KURTZ, AIA

14. JAMES M. LUCKMAN, AIA

The Luckman Partnership, Inc.
Los Angeles, California

15. WILLIAM M. PENA, FAIA

CRSS
Houston, Texas

16. WOLFGANG F. E. PREISER, PhD

PhD in Man-Environment Relations

Architectural Research Consultants, Inc.
Albuquerque, New Mexico

Center for Research and Development
School of Architecture and Planning
University of New Mexico

17. JOHN M. REID

Corporate Planners and Coordinators, Inc.
Arlington, Virginia

18. MICHAEL K. SCHLEY, AIA

FM:Systems
Raleigh, North Carolina

19. FREDERICK J. SCHMIDT

The Environments Group
Chicago, Illinois

20. WALTER H. SOBEL, FAIA

Walter H. Sobel, FAIA and Associates
Chicago, Illinois

Adjunct Professor
Illinois Institute of Technology

21. R. DAVIS WINESETT, JR., AICP, AIA

The Benham Group
Oklahoma City, Oklahoma

Letter Requesting Release of Names

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-6583

REPLY TO ATTN OF: Capt Michael A. Ross (AFIT/GEM/90-S) July 31, 1990

SUBJECT: Air Force Facility Programming and Its Effect on Design and Construction

TO: Mr. Kurt Nuebek

1. I'd like to thank you for participating in my thesis research. I will complete my thesis in a few weeks and I have one final request. I would like to name you as a participant in my research. Your name and professional status will help in establishing the validity of my "expert" panel. Your individual answers and written comments to my research questions, though, will remain anonymous.

2. My thesis should be published in January 1991. The Air Force Institute of Technology controls the distribution and release of thesis information. However, I will aid you in getting a copy of complete document on request.

3. I'm preparing an "executive summary" of the research results because of the time constraints involved with the complete thesis document. The summary will be mailed to you, on request, in late September or early October 1990. In addition, I have to prepare a journal article on my thesis work. This is requirement for one of my current courses. A copy of the article will, also, be mailed to you on request.

4. I have enclosed a short form requesting release of your name, your firm's or institute's name, and your professional or educational status. Also, the form asks you to indicate your interest in copies of the thesis, executive summary, and journal article. Please take a few minutes to complete and mail the form. A pre-addressed, pre-stamped envelop is included.

5. Again, thank you for your help. I realize your time and expertise are valuable. Call me at (513) 236-3241 if you have any questions.

Michael A. Ross

MICHAEL A. ROSS, Capt, USAF
Graduate Engineering Management
Air Force Institute of Technology
School of Systems and Logistics

1 Atch
Release Form

STRENGTH THROUGH KNOWLEDGE

Release Form

RELEASE FORM

Thesis: An Analysis of Air Force Facility Programming and Its Effect on Design and Construction

Student: Michael A. Ross

School: Air Force Institute of Technology
School of Systems and Logistics

Program: Engineering Management

PLEASE ANSWER THE FOLLOWING QUESTIONS:

1. Can I name you as a participant in my thesis work?

_____ YES
_____ NO

2. If so, how would you like your name to appear? Please indicate below.

NAME: ______________________________

3. Would you like your professional or educational status indicated?

_____ YES
_____ NO

4. If so, mark all appropriate blocks below.

_____ FAIA
_____ AIA
_____ PE
_____ PHD, Of What? Please specify below.
____
_____ OTHER, Please specify below.
____

5. Can I name the firm or institute with whom your associated?

_____ YES
_____ NO

6. If so, how would you like the firm's or institute's name to appear? Please indicate below.

FIRM OR
INSTITUTE: ______________________________

1

7. Can I include your, the firm's, or institute's location (city and state only) in my thesis?

_____ YES
_____ NO

8. Would you like a copy of an "Executive Summary" of the results of my thesis?

_____ YES
_____ NO

9. Would you like a copy of the journal article?

_____ YES
_____ NO

10. Would like a copy of the thesis?

_____ YES
_____ NO

PLEASE SIGN AND DATE BELOW.

______________________________ (Signature) ______________ (Date)

2

Appendix B: Round One Delphi Questionnaire Packages

GENERAL INFORMATION

The purpose of the Delphi questionnaires was to gather information on facility programming practices and attitudes. The survey instruments' recipients were two panels of "experts": (1) programming professionals outside the Air Force, and (2) Air Force Chief Engineers at base level Civil Engineering organizations. The two panels were designated Group A and Group B, respectively.

THE QUESTIONNAIRE PACKAGES

The two groups received similar questionnaire packages. The packages included: (1) a cover letter, (2) general instructions, and (3) the questionnaire. Only the Group B package is contained in the appendix. The survey instruments, except for two questions, were the same. The two questions were 5 and 55. Group A's questions were:

5. What type of services do you or your firm provide?

   A. PROGRAMMING, ARCHITECTURAL AND ENGINEERING DESIGN
   B. PROGRAMMING AND ARCHITECTURAL DESIGN
   C. PROGRAMMING AND ENGINEERING DESIGN
   D. PROGRAMMING ONLY
   E. OTHER, PLEASE SPECIFY

55. What two of three questions would you like to ask your peers about facility programming?

In addition, all references to "using/agency" in Group B's questionnaire were changed to "client/user" in the Group A's questionnaire.

Letter to the Group A Participants

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-6583

REPLY TO ATTN OF: Capt Michael A. Ross (AFIT/GEM/DEM/90-S) May 8, 1990

SUBJECT: Air Force Facility Programming and Its Effect on Design and Construction

TO: Mr. Kurt Nuebek

1. Programming is an essential part of facility project management. However, often confusion surrounds programming's interface with building design and construction. I am conducting this study to clearly identify the principal components of successful programming. The information will aid in recommending improvements to the Air Force design and construction process. The construction community, both civilian and military, can benefit through increased customer satisfaction, reduced project costs, and improved construction quality.

2. I am using the "Delphi Method" to research the issues involving the programming process. One of the key features of the "Delphi Method" is the use of experts because of their knowledge and judgment in the research area. As an expert in the field of facility programming, your participation is invaluable to the study's success.

3. Anonymity is another primary feature of the "Delphi Method". The research's success relies on treating all your responses as confidential. In addition, the study will not identify any individuals or organizations unless specific written permission is granted.

4. The "Delphi Method" also is an iterative process. You will receive feedback on the results of each round of questionnaires. In addition, an executive summary of the final results of the research will be mailed to each participant.

5. Again, your input is valuable to improving Air Force facility programming. Please return your responses in the attached, pre-addressed envelop within 7 days of receipt. Call me at (513) 236-3241 if you have any questions about the questionnaire. Thank you for your assistance.

Michael A. Ross

MICHAEL A. ROSS, Capt, USAF
Graduate Engineering Management
Air Force Institute of Technology
School of Systems and Logistics

1 Atch
Survey Packet

STRENGTH THROUGH KNOWLEDGE

Letter to the Group B Participants

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-6583

REPLY TO ATTN OF: Capt Michael A. Ross (AFIT/GEM/DEM/90-S) May 16, 1990

SUBJECT: Air Force Facility Programming and Its Effect on Design and Construction

TO: Chief Engineer

1. Programming is an essential part of facility project management. However, often confusion surrounds programming's interface with building design and construction. I am conducting this study to clearly identify the principal components of successful programming. The information will aid in recommending improvements to the Air Force design and construction process. The construction community, both civilian and military, can benefit through increased customer satisfaction, reduced project costs, and improved construction quality.

2. I am using the "Delphi Method" to research the issues involving the programming process. One of the key features of the "Delphi Method" is the use of experts because of their knowledge and judgment in the research area. As a Chief Engineer your expertise in facility construction is invaluable to the study's success.

3. Anonymity is another primary feature of the "Delphi Method". The research's success relies on treating all your responses as confidential. In addition, the study will not identify any individuals or organizations unless specific written permission is granted.

4. The "Delphi Method" also is an iterative process and will include two rounds of questionnaires. You will receive feedback on the results of each round of questionnaires. In addition, an "Executive Summary" of the final results of the research will be mailed to each participant.

5. Again, your input is valuable to improving Air Force facility programming. Please return your responses in the attached, pre-addressed envelop within 7 days of receipt. Call me at (513) 236-3241 if you have any questions about the questionnaire. Thank you for your assistance.

Michael A. Ross

MICHAEL A. ROSS, Capt, USAF
Graduate Engineering Management
Air Force Institute of Technology
School of Systems and Logistics

1 Atch
Survey Packet

STRENGTH THROUGH KNOWLEDGE

FACILITY PROGRAMMING QUESTIONNAIRE

AFIT SCHOOL OF SYSTEMS AND LOGISTICS
GRADUATE ENGINEERING MANAGEMENT

The purpose of this study is to gather information on the facility programming and its role in the design and construction process.

General Instructions

1. Facility programming, for the purpose of this study, is defined as project definition for construction projects.

2. Please answer each question to the best of your ability. Select only one answer unless directions state otherwise.

3. Circle or mark your answers on the questionnaire. The responses will be calculated by hand, so feel free to comment on any of the questions. Use the back of the sheets when more space is needed.

4. Again, elaborate if you feel an need to qualify an answer or comment on a question. Feedback is an important part of the "Delphi Method", and is appreciated.

5. When you have completed all the items, please put the questionnaire in the envelope provides and send to Capt Michael A. Ross, AFIT/GEM/DEM/90-S, Wright-Patterson AFB, OH 45433-6583. Thank you for your participation.

Round One Questionnaire for Group B

FACILITY PROGRAMMING QUESTIONNAIRE

I. DEMOGRAPHIC QUESTIONS: Questions 1 - 5 ask about your experience and educational background. Please circle the most appropriate answer on the questionnaire. Select only one answer to each question.

1. Your educational background is in:

A. ARCHITECTURE
B. CIVIL ENGINEERING
C. INDUSTRIAL ENGINEERING
D. MECHANICAL ENGINEERING
E. ELECTRICAL ENGINEERING
F. OTHER, PLEASE SPECIFY ____________________

2. How many years of experience do you have in facility programming?

A. NONE
B. LESS THAN 5
C. 5 TO 7
D. 8 TO 10
E. 11 TO 13
F. 14 OR MORE

3. How many years of experience do you have in facility design?

A. NONE
B. LESS THAN 5
C. 5 TO 7
D. 8 TO 10
E. 11 TO 13
F. 14 OR MORE

4. How many years of experience do you have in facility construction management or inspection?

A. NONE
B. LESS THAN 5
C. 5 TO 7
D. 8 TO 10
E. 11 TO 13
F. 14 OR MORE

5. How many years of experience do you have working in Air Force Civil Engineering?

A. LESS THAN 5
B. 5 TO 7
C. 8 TO 10
D. 11 TO 13
E. 14 OR MORE

1

II. RATING SCALES: Questions 6 - 40 ask about your opinions on programming and design issues. A five point rating scale is provided for your responses. The responses range from "strongly agree" to "strongly disagree". Please circle your most appropriate answer on the questionnaire. Select only one answer to each question.

6. Facility programming identifies the functional building requirements for design.

A.__________B.__________C.__________D.__________E.

STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

7. Facility programming identifies the technical building requirements for design.

A.__________B.__________C.__________D.__________E.

8. A facility programming document is a problem definition or statement.

A.__________B.__________C.__________D.__________E.

9. A facility design is a problem solution.

A.__________B.__________C.__________D.__________E.

10. Programming is the responsibility of the user/using agency.

A.__________B.__________C.__________D.__________E.

11. Programming is the responsibility of the designer.

A.__________B.__________C.__________D.__________E.

12. Programming is an iterative process.

A.__________B.__________C.__________D.__________E.

13. Design is an iterative process.

A.__________B.__________C.__________D.__________E.

14. Programming is a series of user/using agency design decisions.

STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

A.__________B.__________C.__________D.__________E.

15. User/using agency participation is very important in programming.

A.__________B.__________C.__________D.__________E.

16. A programmer should guide users/using agencies through decision making.

A.__________B.__________C.__________D.__________E.

17. Users/using agencies should be part of the programming team.

A.__________B.__________C.__________D.__________E.

18. Designers should be part of the programming team.

A.__________B.__________C.__________D.__________E.

19. It is important to educate the users/using agencies in the programming process.

A.__________B.__________C.__________D.__________E.

20. It is important to educate the users/using agencies in architectural design.

A.__________B.__________C.__________D.__________E.

21. Three-way communication between the designer, programmer and user/using agency is essential to programming.

A.__________B.__________C.__________D.__________E.

22. A facility programming document is primarily information for the designer.

A.__________B.__________C.__________D.__________E.

23. A facility programming document is primarily information for the user/using agency.

STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

A.__________B.__________C.__________D.__________E.

24. Conceptual design and contract document production are two separate phases of the design process.

A.__________B.__________C.__________D.__________E.

25. Conceptual design is part of the programming process.

A.__________B.__________C.__________D.__________E.

26. Programming should be completed prior to design.

A.__________B.__________C.__________D.__________E.

27. Programming should be integrated with design.

A.__________B.__________C.__________D.__________E.

28. Programming and design should be interactive, not separate phases of the facility delivery process.

A.__________B.__________C.__________D.__________E.

29. The programming process is the same for all facility projects.

A.__________B.__________C.__________D.__________E.

30. Programming is essential regardless of project size.

A.__________B.__________C.__________D.__________E.

31. The end product of programming is information not design.

A.__________B.__________C.__________D.__________E.

32. A facility programming document should include the quantitative requirements of the user/using agency's organization.

STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

A.__________B.__________C.__________D.__________E.

33. A facility programming document should include the qualitative requirements of the user/using agency's organization.

A.__________B.__________C.__________D.__________E.

34. Programming should always produce a formal document.

A.__________B.__________C.__________D.__________E.

35. A programmer should have experience in design.

A.__________B.__________C.__________D.__________E.

36. A programmer should be competent in communication skills, including graphic analysis and display.

A.__________B.__________C.__________D.__________E.

37. A programmer should understand the whole building delivery process.

A.__________B.__________C.__________D.__________E.

38. During the programming process, uncovering the true needs of the user/using agency is a recurring problem.

A.__________B.__________C.__________D.__________E.

39. During the programming process, getting users/using agencies to make decisions is a recurring problem.

A.__________B.__________C.__________D.__________E.

40. During the facility delivery process, the programming - design relationship/connection is a recurring problem.

A.__________B.__________C.__________D.__________E.

5

III. MULTIPLE CHOICE: Questions 41 - 50 ask for more specific information about programming and design. The answers are given in a multiple choice format. Please circle the most appropriate answer on the questionnaire. Select only one answer to each question.

41. How many opportunities, on the average, do your users/using agencies have to review, verify, change or add to the programming information?

A. 0
B. 1
C. 2
D. 3
E. 4
F. 5 OR MORE

42. How many design solutions, on the average, do you or your A-E firm present the user/using agency?

A. 1
B. 2
C. 3
D. 4
E. 5
F. 6 OR MORE

43. In your opinion, what percentage of overall project development time should be spent on programming.

A. LESS THAN 5%
B. 5% TO 10%
C. 11% TO 15%
D. 11% TO 20%
E. 20% TO 25%
F. 26% OR MORE

44. You or your firm use a computer (not including word processing) to perform:

A. MOST
B. SOME
C. LITTLE
D. NONE

of the analyzing, organizing and evaluating of programming data.

45. Programming is:

A. PART OF THE DESIGN PROCESS
B. SEPARATE FROM THE DESIGN PROCESS

46. Conceptual design is:

A. PART OF THE PROGRAMMING PROCESS
B. PART OF THE DESIGN PROCESS
C. PART OF BOTH THE DESIGN AND PROGRAMMING PROCESSES
D. SEPARATE FROM THE DESIGN AND PROGRAMMING PROCESSES
E. OTHER, PLEASE SPECIFY BELOW

____

47. The distinct phases of the facility delivery process are:

A. PROGRAMMING, CONCEPTUAL DESIGN, DESIGN (contract documents), and CONSTRUCTION
B. PROGRAMMING, DESIGN, and CONSTRUCTION
C. DESIGN and CONSTRUCTION
D. OTHER, PLEASE SPECIFY BELOW

____

48. In your opinion, who should control the programming of facility projects.

A. USER/USING AGENCY
B. DESIGNER OR DESIGN TEAM
C. IN-HOUSE PROGRAMMING STAFF
D. OUTSIDE PROGRAMMING CONSULTANTS (A-E firms)
E. OTHER, PLEASE SPECIFY BELOW

____

49. Programming includes:

A. DETAILS FOR CONTRACT DOCUMENTS PRODUCTION.
B. MAJOR ISSUES FOR CONCEPTUAL DESIGN.
C. BOTH

50. The Architect's Guide to Facility Programming lists three basic approaches to programming, which of the following approaches best describes your programming method.

A. SEGREGATED: Programming is separate, distinct activity (1) performed prior to initiating of designing, and (2) performed by a separate individuals or teams from the designers.

B. INTEGRATED: Programming is not a "predesign" service, but an integral first part of the design process.

C. INTERACTIVE: Programming and designing are performed in alternating sequence and in response to each other.

IV. MULTIPLE CHOICE: Questions 51 - 54 ask for more specific information about programming content and methods. The answers are given in a multiple choice format. Please mark all appropriate answers on the questionnaire.

51. A programming document almost always should include:

___ Organizational Goals and Objectives (Using Agency)
___ Functional Requirements
___ Technical Requirements
___ Budget and Cost Information
___ Schedule Information
___ Environmental Data
___ Energy Requirements
___ Other, Please Specify Below
1.
2.
3.

52. Which of the following techniques have you used when COLLECTING programming information.

___ Background Data Research
___ Surveys
___ Interviews
___ Questionnaires
___ Data Logs
___ Standardized Data Forms
___ Direct Observation
___ Tracking
___ Participant Observation
___ Behavior Mapping
___ Behavior Specimen Record
___ Instrumented Observation
___ Semantic Differential
___ Adjective Checklist
___ Attribute Discrimination Scale
___ Ranking Chart
___ Preference Matrix
___ Others, Please Specify Below
1.
2.
3.

53. Which of the following techniques have you used for ANALYZING and ORGANIZING programming data?

___ Descriptive Statistics
___ Inferential Statistics
___ Behavior Setting Survey
___ Activity Site Model
___ Time Budget Analysis
___ Pattern Language
___ Space Unit Standards
___ Space Program
___ Energy Budgeting
___ Project Cost Estimating
___ Construction Cost Estimating
___ Life Cycle Cost Analysis
___ Value Analysis
___ Cost-Benefit Analysis
___ Bar Chart/Milestone Chart
___ Activity Time Chart
___ Critical Path Method (CPM)
___ Program Evaluation and Review Technique (PERT)
___ Precedence Diagraming Method (PDM)
___ Relationship Matrices
___ Social Map
___ Sociogram
___ Behavior Map
___ Bubble Diagram
___ Link-Mode Diagram
___ Block Diagram
___ Interaction Net
___ Dual Graph
___ Adjacency Diagram
___ Functional Relationship Diagram
___ Layout Diagram
___ Flow Diagram
___ Organizational Chart
___ Analysis Cards
___ Worksheets
___ Others, Please Specify Below

1.
2.
3.

54. Which of the following techniques have you used for COMMUNICATING and EVALUATING programming data?

___ Brainstorming
___ Synetics
___ Buzz/Rap Session
___ Role Playing
___ Gaming
___ Group Planning
___ Narrative
___ Graphics
___ Audio/Visual Aids
___ Oral Presentations
___ Forums
___ Panel Discussions
___ Work/Charrette/Primer Books
___ Rating and Rating Scales
___ Ladder Scale
___ Rating Chart
___ Evaluation Matrix
___ Weighting
___ Others, Please Specify Below
1.
2.
3.

V. OPEN-ENDED QUESTION: Question 55 is asked to solicit your concerns about facility programming. Again thank you for your help.

55. Do you believe Air Force programming methods adequately define project requirements prior to initiating design? Please explain.

Appendix C: Round One Delphi Questionnaire Packages

GENERAL INFORMATION

The purpose of the Delphi questionnaires was to gather information on facility programming practices and attitudes. The second round survey instruments expanded on the round one questionnaires. The goal of the Delphi technique is to reach consensus on an issue or question. The round two questionnaires consisted of round one questions that did not meet the consensus criteria. In the second round, these questions were reexamined. Again, the recipients were two panels of "experts": (1) programming professionals outside the Air Force, and (2) Air Force Chief Engineers at base level Civil Engineering organizations. The two panels were designated Group A and Group B, respectively.

THE QUESTIONNAIRE PACKAGES

The two groups received similar questionnaire packages. The packages included: (1) a cover letter, (2) general instructions, (3) instructions on "How to Read the Questionnaire", and (4) the questionnaire. The round two questionnaires contained statistical data and written responses from the first round. The descriptive statistics included: (1) the frequency of each response, (2) the percentage of each response, (3) the number of respondents, (4) the mean response, and (5) the median response.

However, the Group A and Group B survey instruments were quite different. The participants received just the data from their group. In addition, consensus was determined separately for each panel of "experts, establishing the questions for that group. Since Group A and Group B each received a different mix of questions and data, both questionnaires are included in the appendix.

Letter to the Group A Participants

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-6583

REPLY TO ATTN OF: Capt Michael A. Ross (AFIT/GEM/90-S) June 15, 1990

SUBJECT: Air Force Facility Programming and Its Effect on Design and Construction

TO: Mr. Kurt Nuebek

1. Thank you for your participation in the first round questionnaire on facility programming. The second survey includes 25 of the original 50 questions for reexamination. The other 25 questions were eliminated because (1) consensus was reached on the first round, or (2) the question dealt with purely demographic data.

2. Respondents: I received 20 questionnaires from my original panel of 25 "experts." The group of participants was fairly homogeneous. The respondents: (1) all have backgrounds in architecture, and (2) all work for firms or institutions that provide programming as a service. In addition, 85 percent of the participants have 10 or more years of programming experience.

3. Consensus: The main objective of my research method, the Delphi Technique, is the consensus of participants on an issue or question. For the purposes of this study, the criteria for consensus for multiple choice and rated scale questions is:

a. Multiple Choice - A 60 percent agreement among respondents on a single answer constitutes consensus for multiple choice questions.

b. Rated Scale - A 70 percent agreement among respondents on rated scaled questions constitutes consensus based on two groups of responses: "strongly agree/agree" and "strongly disagree/disagree."

4. Again, your input is valuable to improving Air Force facility programming. Please return your responses to the second questionnaire in the enclosed, pre-addressed envelop as soon as possible. Call me at (513) 236-3241 if you have any questions about the questionnaire. Thank you for your assistance.

Michael A. Ross

MICHAEL A. ROSS, Capt, USAF
Graduate Engineering Management
Air Force Institute of Technology
School of Systems and Logistics

1 Atch
Survey Packet

STRENGTH THROUGH KNOWLEDGE

Letter to the Group B Participants

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-6583

REPLY TO ATTN OF: Capt Michael A. Ross (AFIT/GEM/90-S) June 22, 1990

SUBJECT: Air Force Facility Programming and Its Effect on Design and Construction

TO: Chief Engineer

1. Thank you for your participation in the first round questionnaire on facility programming. The second survey includes 27 of the original 51 questions for reexamination. The other 24 questions were eliminated because (1) consensus was reached on the first round, or (2) the question dealt with purely demographic data.

2. Respondents: I received 31 questionnaires from my original panel of 40 "experts." The participants are all Chief Engineers at Air Force bases in the CONUS. In addition, 77 percent of the participants have 10 or more years of experience working in Air Force Civil Engineering organizations.

3. Consensus: The main objective of my research method, the Delphi Technique, is the consensus of participants on an issue or question. For the purposes of this study, the criteria for consensus for multiple choice and rated scale questions is:

a. Multiple Choice - A 60 percent agreement among respondents on a single answer constitutes consensus for multiple choice questions.

b. Rated Scale - A 70 percent agreement among respondents on rated scaled questions constitutes consensus based on two groups of responses: "strongly agree/agree" and "strongly disagree/disagree."

4. Again, your input is valuable to improving Air Force facility programming. Please return your responses to the second questionnaire in the enclosed envelop as soon as possible. Call me at (513) 236-3241 if you have any questions about the questionnaire. Thank you for your assistance.

Michael A. Ross

MICHAEL A. ROSS, Capt, USAF
Graduate Engineering Management
Air Force Institute of Technology
School of Systems and Logistics

1 Atch
Survey Packet

STRENGTH THROUGH KNOWLEDGE

FACILITY PROGRAMMING QUESTIONNAIRE

AFIT SCHOOL OF SYSTEMS AND LOGISTICS
GRADUATE ENGINEERING MANAGEMENT

The purpose of this study is to gather information on the facility programming and its role in the design and construction process.

General Instructions

1. Please read "How To Read the Questionnaire" (enclosed) before attempting to answer the survey. Answer each question to the best of your ability. For example, if you are an educator use your past experience. Select only one answer unless directions state otherwise.

2. Circle or mark your answers on the questionnaire. The responses will be calculated by hand, so feel free to comment on any of the questions. Use the back of the sheets when more space is needed.

3. Again, elaborate if you feel an need to qualify an answer or comment on a question. Feedback is an important part of the "Delphi Method", and is appreciated.

4. When you have completed all the items, please put the questionnaire in the pre-stamped envelope provided and send to Capt Michael A. Ross, AFIT/GEM/90-S, Wright-Patterson AFB, OH 45433-6583. Thank you for your participation.

Instructions on "How to Read the Questionnare"

HOW TO READ THE QUESTIONNAIRE

A. FORMAT: The second round questionnaire is broken into five sections. Each section contains 3 to 6 related questions with the appropriate data from the first questionnaire. The data is included to give insight into how the other "experts" feel about a particular subject or question. Please give consideration to this information when responding to the questions.

B. COMMENTS: Each section contains written comments by the respondents from the first questionnaire. Please read the comments for they are valuable source of information. The bracketed text (i.e. [programming is]) was added to clarify the response as it pertained to the question.

C. RATING SCALES: The questions with responses on a five point rating scale were evaluated by giving a numerical value to each response as follows:

5 = STRONGLY AGREE
4 = AGREE
3 = UNDECIDED
2 = DISAGREE
1 = STRONGLY DISAGREE

The second round questions include data from the first questionnaire. Each question includes the following data: (1) the frequency of each response, (2) the percentage of each response, (3) the number of responses, (4) the *mean* (or average) response, and (5) the median (or middle) response. (See example below.)

Example:

21. Three-way communication between the designer, programmer and client is essential to programming.

Frequency (number) of responses to 'B' - AGREE

(14)	(3)	(1)	(1)	(1)
A.	B.	C.	D.	E.
(70%)	(15%)	(5%)	(5%)	(5%)

Percentage of total responses to 'D' - DISAGREE

N = 20 **MEAN = 3.650** **MEDIAN = 4.000**

Number of total responses to question

1

D. MULTIPLE CHOICE: The multiple choice questions include the following data from the first questionnaire: (1) *frequency* of each response and (2) percentage of each response. (See example below.)

<u>**Example:**</u>

49. Programming includes:

First column is frequency (number) of responses to each answer.

(0)	(0%)	A.	DETAILS FOR CONTRACT DOCUMENTS PRODUCTION
(13)	(65%)	B.	MAJOR ISSUES FOR CONCEPTUAL DESIGN
(7)	(34%)	C.	BOTH

Second column is percentage of total responses to each answer.

E. CHANGES AND CLARIFICATIONS: In response to respondent comments, the second round questionnaire includes additions, omissions and definitions of words or phrases contained in particular questions.

(1) <u>Additions</u>: Words or phrases added to a question are italicized. For example, the word "users" was added to the following question.

19. It is important to educate clients/*users* in the programming process.

(2) <u>Omissions</u>: Words or phrases omitted from the question, but were included in the first questionnaire are bracketed. For example, the word "always" was originally part of the following question, but should not be considered now. The word is included only to give context to answers given in the original version of the question.

34. Programming should [always] produce a formal document.

(3) <u>Definitions</u>: Words or phrases that are defined in each section are bold-faced. For example, the word "iterative" in the following question would be defined in the section under "Definitions." Definitions are included, in some cases, to clarify the meaning of certain words for a particular question.

12. Programming is an **iterative** process.

Round Two Questionnaire for Group A

FACILITY PROGRAMMING QUESTIONNAIRE - ROUND 2

SECTION 1: Questions 10, 11, 14, 22, 23 and 48 deal with the roles of the client and designer in programming. Please: (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

A.	B.	C.	D.	E.
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

10. Programming is the responsibility of the client/owner.

A.	B.	C.	D.	E.
(9)	(4)	(1)	(3)	(3)
(45%)	(20%)	(5%)	(15%)	(15%)

N = 20 MEAN = 3.650 MEDIAN = 4.000

11. Programming is the responsibility of the designer.

A.	B.	C.	D.	E.
(3)	(5)	(3)	(2)	(6)
(16%)	(26%)	(16%)	(10%)	(32%)

N = 19 MEAN = 2.842 MEDIAN = 3.000

14. Programming is a series of client [design] decisions on the direction of design.

A.	B.	C.	D.	E.
(4)	(5)	(0)	(6)	(5)
(20%)	(25%)	(0%)	(30%)	(25%)

N = 20 MEAN = 2.800 MEDIAN = 2.000

22. A facility programming document is primarily information for the designer.

A.	B.	C.	D.	E.
(6)	(3)	(3)	(8)	(0)
(30%)	(15%)	(15%)	(40%)	(0%)

N = 20 MEAN = 3.300 MEDIAN = 3.000

23. A facility programming document is [primarily] valuable information for the client.

A.	B.	C.	D.	E.
(4)	(4)	(3)	(8)	(1)
(20%)	(20%)	(15%)	(40%)	(5%)

N = 20 MEAN = 3.650 MEDIAN = 4.000

1

48. In your opinion, who should control the programming process of facility projects. (Assume client/owner has no in-house capability and design firm has in-house programming staff)

(4)	(21%)	A.	CLIENT/OWNER
(2)	(10.5%)	B.	DESIGNER OR DESIGN TEAM
(5)	(26%)	C.	IN-HOUSE PROGRAMMING STAFF (part of design firm)
(3)	(15.5%)	D.	OUTSIDE PROGRAMMING CONSULTANTS (separate from design firm)
(5)	(26%)	E.	OTHER (see below)

1. Varies by project conditions, project type, and designer/owner expertise

2. Client always has responsibility for decision, design team should usually be responsible for the programming process

3. 'C' or 'D' - small firms cannot always have in house staff; they might require consultation

4. Client controls; [but] can't generalize, varies by type/competence of clients.

5. 'C' is best but only if qualified programming professionals are the design staff, which is rare. If not, 'D' is best. In other words, quality is most important, then integration with design team; preferably both are provided.

__Definitions:__

responsibility - the state of being liable or accountable as the primary agent

control - directing influence over

__Comments:__

__Questions 10 and 11__

- Program is client responsibility, but programming is architect's responsibility.
- Yes [client responsibility], if client can do it.
- Programmer assists client/owner in making decisions which is their responsibility.
- In reality, it [programming] becomes the responsibility of the owner. It should be the responsibility of the designer.
- Most owners cannot develop a complete one [program].
- Yes [designer responsibility], if retained to do it [programming].
- Architect confirms the requirements of project to the owner as the first step in basic services.
- [Programming is] responsibility of the programmer or designer.
- The designer should participate or interface with the programmer for a complete project.

- [Programming is] team effort with client input; shared responsibility.

Question 14

- Decisions yes; design decisions, no.
- Client input is important; there are no decisions, there are guidelines.
- Programming sets direction/concepts for design and eliminates options.

Questions 22 and 23

- It [programming document] has important uses for client.
- [Programming document] both [for] designer and client. Client needs to have their expectations articulated so they can participate knowledgeably in the design decision process.
- Document is feedback to client for approval.
- [Programming document] is equally important to designer.
- It [programming document] is often a commitment document of the staff and CEO, if it is used as a sign-off document. In the case of an A/E [Architect/Engineer] contract, the program is a contract document on which fees and project costs are based.
- [Programming document is] primarily for designer but also a valuable reference for client.
- The client is responsible for approving the program. The document is his contract with the designer.
- A sophisticated client will use the programming document as a management tool and for future requirements database; also a critical document for design of the facility.

Question 48

- [Client controls], but [designer, in-house staff, and outside consultants] should have strong influence
- Programmer should be either on staff of client/owner or a direct consultant to them. Client controls.

END OF SECTION 1

SECTION 2: Questions 18, 25, 35, 37, 46 and 47 deal with the Programming/Conceptual (schematic) Design interface. Please: (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

A.	B.	C.	D.	E.
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

18. Designers should be part of the programming team.

(7)	(7)	(3)	(1)	(2)
A.	B.	C.	D.	E.
(35%)	(35%)	(15%)	(5%)	(10%)

N = 20 MEAN = 3.750 MEDIAN = 4.000

25. Conceptual design is part of the programming process.

(3)	(3)	(4)	(7)	(3)
A.	B.	C.	D.	E.
(15%)	(15%)	(20%)	(35%)	(15%)

N = 20 MEAN = 2.800 MEDIAN = 2.500

35. A programmer or someone on the programming team should have experience in design.

(7)	(7)	(4)	(2)	(0)
A.	B.	C.	D.	E.
(35%)	(35%)	(20%)	(10%)	(0%)

N = 20 MEAN = 3.950 MEDIAN = 4.000

37. A programmer or someone on the programming team should understand the whole building delivery process.

(9)	(5)	(4)	(1)	(1)
A.	B.	C.	D.	E.
(45%)	(25%)	(20%)	(5%)	(5%)

N = 20 MEAN = 4.000 MEDIAN = 4.000

46. Conceptual design is:

(4)	(20%)	A.	PART OF THE PROGRAMMING PROCESS
(9)	(45%)	B.	PART OF THE DESIGN PROCESS
(5)	(25%)	C.	PART OF BOTH THE DESIGN AND PROGRAMMING PROCESSES
(2)	(10%)	D.	SEPARATE FROM THE DESIGN AND PROGRAMMING PROCESSES
(0)	(0%)	E.	OTHER

47. The distinct phases of the facility delivery process are:

(13)	(65%)	A.	PROGRAMMING, CONCEPTUAL DESIGN, DESIGN, and CONSTRUCTION
(2)	(10%)	B.	PROGRAMMING, DESIGN, and CONSTRUCTION
(0)	(0%)	C.	DESIGN and CONSTRUCTION
(5)	(25%)	D.	OTHER (See below)

1. Programming, Conceptual Design, and Contract Documents are interactive
2. Strategic Planning (needs assessment/project identification), Programming, Design, Construction, Activation, Evaluation, Retirement
3. Phase I - Programming, Concept Design, Budget, Schedule; Phase II - Design Development/Contract Documents; Phase III - Bid, Construction Award
4. I use an iterative approach with overlapping programming phases. Our design phases are Concept, Schematic, Design Development, Construction Documentation.
5. Programming is a concurrent activity which starts ahead of the design process and continues until after move-in.

<u>Definitions:</u>

Conceptual Design - means conceptual or Schematic Design per A.I.A standard terms.

Design - means Design Development and Contract Document production per A.I.A. standards.

<u>Comments:</u>

<u>Question 18</u>

- [Programmers should not be part of the programming team] unless they can be objective.
- Depends, [designers as part of the programming team] can work very well (or not).
- [Designers should be part of the programming team] as observers and contributors.
- A clear program should allow designers to proceed.
- [Designers as part of the programming team], a must for effective follow through - the designer always finishes the program in practice.

<u>Question 25</u>

- [Conceptual design] adjusts the program.
- It [programming process] is iterative.
- [Conceptual design is not part of the programming process] but [it] can be valuable to explore design implications.
- Depends on the client and schedule.

- They [conceptual design and programming] can be mutually supportive and time saving to do coordination with schematics, except that detailed technical info (data sheets) can wait.
- Preferably [conceptual design is part of the programming process], but sometimes i. separated successfully.

Question 35

- [A programmer should have experience in design] at least in school.
- [A programmer with design experience is] desirable, but not critical.
- Design experience certainly helpful, but I have trained people without design experience to be excellent programmers.
- Depends; someone on [programming] team should [have experience in design.
- [A programmer with design experience is] helpful but not essential.
- It [a programmer with design experience] is very helpful but not essential.
- [A programmer should be] sensitive to design but not mandatory to have experience in design.

Question 37

- [A programmer should understand the building delivery process] at some level.
- [A programmer should understand the building delivery process] - not necessarily.
- This [understanding the building delivery process] certainly helps. I've seen many owner[s] who have retained prominent number crunching firms or planning-only firms who have produced incredibly poor and understated programs and budgets, including the military. Then a knowledgeable A/E is brought on board, and he has to begin a business and personal relationship by telling the owner a lot of bad news, or risks being liable for the eventual consequences. Then the budget is upset while programming is revised and additional funding secured, while time flies by and escalations erodes the budget further.
- Someone on [programming] team should [understand the building delivery process].
- A general understanding [of the building delivery process] is needed.

Questions 46 and 47

- [Conceptual design is part of the design process] but best shared during programming.
- [Programming and conceptual design are] joint activities; [Design] should be schematic, design development, contract documents.

END OF SECTION 2

SECTION 3: Questions 26, 27, 28, 29, and 50 deal with the basic approach to programming and design. Please: (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

50. The Architect's Guide to Facility Programming lists three basic approaches to programming, which of the following approaches best describes your programming method.

(8) (40%) A. SEGREGATED: Programming is separate, distinct activity (1) performed prior to initiating of designing, and (2) performed by a separate individuals or teams from the designers.

(6) (30%) B. INTEGRATED: Programming is not a "predesign" service, but an integral first part of the design process.

(4) (20%) C. INTERACTIVE: Programming and designing are performed in alternating sequence and in response to each other.

(2) (10%) D. SEGREGATED or INTERACTIVE, depending on the project.

A.	B.	C.	D.	E.
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

26. Programming should be completed prior to design.

A.	B.	C.	D.	E.
(9)	(4)	(3)	(3)	(1)
(45%)	(20%)	(15%)	(15%)	(5%)

N = 20 MEAN = 3.850 MEDIAN = 4.000

27. Programming should be integrated with conceptual design.

A.	B.	C.	D.	E.
(2)	(7)	(4)	(6)	(1)
(10%)	(35%)	(20%)	(30%)	(5%)

N = 20 MEAN = 3.150 MEDIAN = 3.000

28. Programming and conceptual design should be interactive, not separate phases of the facility delivery process.

(4)	(7)	(3)	(5)	(1)
A.	B.	C.	D.	E.
(20%)	(35%)	(15%)	(25%)	(5%)

N = 20 MEAN = 3.400 MEDIAN = 4.000

29. The programming process is the same for all facility projects.

(2)	(3)	(1)	(8)	(6)
A.	B.	C.	D.	E.
(10%)	(15%)	(5%)	(40%)	(30%)

N = 20 MEAN = 2.350 MEDIAN = 2.000

__Definitions:__

Conceptual Design - means conceptual or Schematic Design per A.I.A standard terms.

Design - means Design Development and Contract Document production per A.I.A. standards.

__Comments:__

__Questions 26, 27, 28, 29 and 30__

- [Yes programming should be completed prior to design] except to evolve into concepts.
- Yes [programming should be completed prior to design], but not always possible or desirable; design often tests program assumptions or requirements.
- A [programming] document [should be completed prior to design.
- Depends; we now (sometimes) do schematic level program [first]; then [complete] detailed program while design is underway. [There] are, however, risks (that change [requirements] as get into detail).
- Client must determine scope [of programming].
- Depends on client and schedule [if programming is completed prior to design].
- [Programming] must be [integrated] early in design process.
- Subject (content) should be integrated not the process.

- No [programming should not be integrated with design], it is an iterative process with design.
- I'd say [programming should be] coordinated [with design not integrated].

- [Programming should] continue concurrently [with design].
- [Programming should be] integrated with conceptual design, but some programming is required to do it.
- Only concepts [should be integrated with design].
- Trial designs should be tested during the programming process to verify interrelationships, and net to gross factors.
- Base building program should proceed schematic and design development.
- [Programming is interactive] with concepts.
- [Programming process is] similar, not the same.
- Nothing is the same [programming process] for all projects.
- [The programming process is] never [the same]; format similar, technique[s] not standard.
- Variations in client terms, data availability, schedule milestones (financial, marketing, PR, ect.) can all influence [the programming] process for each project.
- [Programming is integrated] but include [interactive]; programming is the first step in the process, but often becomes interactive as it is tested during early design stages.
- Normally, when a programming phase is completed [without] any design input, it is made clear that some adjustments will be made in the schematic design phase.
- [The programming method is either segregated or interactive]; depends on the job/project.

END OF SECTION 3

SECTION 4: Questions 9, 20, 31, 40, and 45 are loosely related dealing with the role design in the programming phase. Please: (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

A.	B.	C.	D.	E.
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

9. A facility design is a problem solution.

(11)	(2)	(1)	(1)	(3)
A.	B.	C.	D.	E.
(61%)	(11%)	(5.5%)	(5.5%)	(17%)

N = 18 MEAN = 3.944 MEDIAN = 5.000

20. It is important to educate the client/users in architectural design process.

(7)	(8)	(4)	(1)	(0)
A.	B.	C.	D.	E.
(35%)	(40%)	(20%)	(5%)	(0%)

N = 20 MEAN = 4.000 MEDIAN = 4.000

31. The end product of programming is information not design.

(11)	(3)	(2)	(3)	(1)
A.	B.	C.	D.	E.
(55%)	(15%)	(10%)	(15%)	(5%)

N = 20 MEAN = 4.000 MEDIAN = 5.000

40. During the facility delivery process, the programming - design relationship/connection [is a recurring] can be a problem.

(1)	(11)	(2)	(6)	(0)
A.	B.	C.	D.	E.
(5%)	(55%)	(10%)	(30%)	(0%)

N = 20 MEAN = 3.350 MEDIAN = 4.000

45 *Programming is:*

(9) (45%) A. PART OF THE DESIGN PROCESS

(9) (45%) B. SEPARATE FROM THE DESIGN PROCESS

(2) (10%) C. *INTERACTIVE WITH DESIGN PROCESS*

Comments:

Question 20

- [The] point is to lead client thru the process of design.
- [It is important to educate the client/users in architectural design] process and levels of information development.
- [Educating clients/users in design] depends on the client and their experience. Also, [design] is not really necessarily part of the programming phase.
- Most clients need to understand design approach or design direction to give meaningful data to the programmer.

Question 31

- Many programs really drive design.
- We say that the program is the roadm: to design.
- Depends on the project [if programming is information not design].
- [The end product of programming is information not design] except that concept design does: (a) confirm program concepts and (b) sets firm design directions.
- If not conceptual design, a certain amount of design guidelines and design criteria must be included in programming.

Question 40

- Problem [referring to programming/design connec ion] is architects and other professionals who have been educated to believe that they know best! and do not know how to listen.
- [Programming/design connection is a problem] only if the client abandons the program as the formal embodiment of their needs, or if was a lousy program to begin with.
- [If the programming/design connection is a problem] depends on: (a) designer involvement in program, (b) implementation of the program in its detail, (c) stability of the program related to client changes.
- [The programming/design connection] should not be a problem; integrated team is best.

END OF SECTION 4

SECTION 5: Questions 52, 53, and 54 ask which programming techniques are most effective for collecting, analyzing, organizing, communicating and evaluating data. Only techniques in which 50% or more of respondents marked as "have used" were included in the Round 2 questionnaire.

52. Which of the following techniques [have you used] are most effective when COLLECTING programming information.

(18)	(90%)	___	Background Data Research
(20)	(100%)	___	Surveys
(20)	(100%)	___	Interviews
(20)	(100%)	___	Questionnaires
(12)	(60%)	___	Data Logs
(16)	(80%)	___	Standardized Data Forms
(20)	(100%)	___	Direct Observation
(13)	(65%)	___	Participant Observation
(10)	(50%)	___	Ranking Chart
(14)	(70%)	___	Preference Matrix
		___	Others, Please Specify Below

1.
2.
3.

53. Which of the following techniques [have you used] are most effective for ANALYZING and ORGANIZING programming data?

(17)	(85%)	___	Descriptive Statistics
(11)	(55%)	___	Inferential Statistics
(19)	(95%)	___	Space Unit Standards
(20)	(100%)	___	Space Program
(15)	(75%)	___	Project Cost Estimating
(14)	(70%)	___	Construction Cost Estimating
(13)	(65%)	___	Life Cycle Cost Analysis
(16)	(80%)	___	Bar Chart/Milestone Chart
(11)	(55%)	___	Activity Time Chart
(12)	(60%)	___	Critical Path Method (CPM)
(18)	(90%)	___	Relationship Matrices
(19)	(95%)	___	Bubble Diagram
(13)	(65%)	___	Block Diagram
(19)	(95%)	___	Adjacency Diagram
(18)	(90%)	___	Functional Relationship Diagram
(12)	(60%)	___	Layout Diagram
(18)	(90%)	___	Flow Diagram
(18)	(90%)	___	Organizational Chart
(10)	(50%)	___	Worksheets
		___	Others, Please Specify Below

1.
2.
3.

12

54. Which of the following techniques [have you used] are most effective for COMMUNICATING and EVALUATING programming data?

(13)	(65%)	___ Brainstorming
(11)	(55%)	___ Group Planning
(17)	(85%)	___ Narrative
(20)	(100%)	___ Graphics
(19)	(95%)	___ Audio/Visual Aids
(19)	(95%)	___ Oral Presentations
(10)	(50%)	___ Forums
(10)	(50%)	___ Rating and Rating Scales
(11)	(55%)	___ Evaluation Matrix
(11)	(55%)	___ Weighting
		___ Others, Please Specify Below

1.
2.
3.

END OF SECTION 5

THANK YOU FOR
YOUR PARTICIPATION.

Round Two Questionnaire for Group B

FACILITY PROGRAMMING QUESTIONNAIRE - ROUND 2

SECTION 1: Questions 10, 14, 18, 20, 21, and 35 deal with the roles of the user, programmer and designer in programming. Please: (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

A.	B.	C.	D.	E.
(5)	(4)	(3)	(2)	(1)
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

10. Programming is the responsibility of the user/using agency.

(2)	(10)	(2)	(13)	(4)
A.	B.	C.	D.	E.
(6%)	(32%)	(6%)	(42%)	(13%)

N = 31 MEAN = 2.774 MEDIAN = 2.000

14. Programming is a series of user/using agency [design] decisions on the direction of design.

(0)	(12)	(3)	(11)	(5)
A.	B.	C.	D.	E.
(0%)	(39%)	(10%)	(35%)	(16%)

N = 31 MEAN = 2.710 MEDIAN = 2.000

18. Designers should be part of the programming team.

(9)	(14)	(5)	(3)	(0)
A.	B.	C.	D.	E.
(29%)	(45%)	(16%)	(10%)	(0%)

N = 31 MEAN = 3.935 MEDIAN = 4.000

20. It is important to educate the users/using agencies in architectural design process.

(1)	(9)	(8)	(11)	(2)
A.	B.	C.	D.	E.
(3%)	(29%)	(26%)	(35%)	(6%)

N = 31 MEAN = 2.871 MEDIAN = 3.000

21. Three-way communication between the designer, programmer and user/using agency is essential to programming.

(17)	(6)	(3)	(5)	(0)
A.	B.	C.	D.	E.
(55%)	(20%)	(10%)	(16%)	(0%)

N = 31 MEAN = 4.129 MEDIAN = 5.000

1

35. A programmer or someone on the programming team should have experience in design.

(5)	(14)	(5)	(4)	(1)
A.	B.	C.	D.	E.
(17%)	(48%)	(17%)	(14%)	(3%)

N = 31 MEAN = 3.621 MEDIAN = 4.000

Definitions:

responsibility - the state of being liable or accountable as the primary agent

Comments:

Question 10

- But if they [the user] don't get serious at the programming stage, much time and money is wasted later in the process.
- Knowing what they [the user] need and when - yes; documents - no.
- User identifies requirement/problem - CE is responsible for programming using user input.
- Designer should solve the users' problem, not just design what the user wants or think he wants.

Question 14

- Actually, we [Civil Engineering] make the decisions. The user defines the problem.
- [Programming is] identifying user requirements and criteria.
- User should not be making design decisions.

Question 18

- But designers are not available to do this [be part of programming team.]
- [Designers should be part of the programming team] for information and input. However some designers become worried over funding and block design decisions.

Question 20

- [Users should be educated in architectural design] only if they insist on defining a solution which is architecturally incorrect.
- [Users should be educated in architectural design to] provide an understanding and as part of the team.
- They [users] need to be aware of architectural compatibility.

Question 21

- Again lack of designer time is a problem here.
- The programmer is the designer under your definition in the general instructions.

Question 35

- Sure would help [experience in design.]
- [The programmer] needs to be able to understand his function vs. design.
- [Experience in design is] beneficial but not mandatory. Hard to find designers that are willing to be programmers.
- [A programmer needs experience in design but] doesn't need much though.
- It [experience in design] would be nice but not necessary.
- Ideally [a programmer should have experience in design.]
- [A programmer] should be the designer not a 1391 writer.

END OF SECTION 1

SECTION 2: Questions 12, 13, 25, 40, 45, and 47 deal with the Programming/Conceptual (schematic) Design interface. Please: (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

A.	B.	C.	D.	E.
(5)	(4)	(3)	(2)	(1)
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

12. Programming is an iterative process.

(5)	(12)	(5)	(8)	(1)
A.	B.	C.	D.	E.
(16%)	(39%)	(16%)	(26%)	(3%)

N = 31 MEAN = 3.387 MEDIAN = 4.000

13. Design is an iterative process.

(3)	(14)	(4)	(7)	(2)
A.	B.	C.	D.	E.
(10%)	(47%)	(13%)	(23%)	(7%)

N = 30 MEAN = 3.300 MEDIAN = 4.000

25. Conceptual design is part of the programming process.

(4)	(14)	(6)	(6)	(0)
A.	B.	C.	D.	E.
(13%)	(47%)	(20%)	(20%)	(0%)

N = 30 MEAN = 3.533 MEDIAN = 4.000

40. During the facility delivery process, the programming - design relationship/connection [is a recurring] can be a problem.

(2)	(11)	(4)	(14)	(0)
A.	B.	C.	D.	E.
(6%)	(35%)	(13%)	(45%)	(0%)

N = 31 MEAN = 3.032 MEDIAN = 3.000

47. The distinct phases of the facility delivery process are:

(17)	(55%)	A.	PROGRAMMING, CONCEPTUAL DESIGN, DESIGN, and CONSTRUCTION
(12)	(39%)	B.	PROGRAMMING, DESIGN, and CONSTRUCTION
(1)	(3%)	C.	DESIGN and CONSTRUCTION
(1)	(3%)	D.	OTHER

1.Could be 'A' or 'B' depending on your desires. I'm not sure which is best. Except that we get poor programming documents with the system described in 'A'.

45. Programming is:

(13) (50%) A. PART OF THE DESIGN PROCESS
(13) (50%) B. SEPARATE FROM THE DESIGN PROCESS

Definitions:

iterative - involving repetition, (i.e. programming information may be presented 2 or 3 times to the using agency for confirmation and approval.)

Conceptual Design - means conceptual or Schematic Design per A.I.A standard terms.

Design - means Design Development and Contract Document production per A.I.A. standards.

Comments:

Question 12

- [No, programming is not an iterative process] unless you are referring to 'how' to fill out the paperwork.

Question 13

- Hopefully [design is] not [an iterative process] for a given problem design. As a process yes.

Question 25

- [Conceptual design] may be desirable to be [part of the programming process] but not possible with current manning.
- Is done both ways [conceptual design either as part or separate from programming process.] The better the programming document, the fewer surprises in the concept design.
- Sometimes [conceptual design is part of the programming process], but not often.
- [Conceptual design is part of the programming process] for MILCON program. [Conceptual design is separate from programming process] for O&M program.

Question 40

- Remembering to keep programmer informed of changes in concepts [is a problem.]
- Depends on program [if design/programming connection is a problem.] On some projects. Usually when new requirements are identified 9user, mission changes, regulations) or funding was approved prior to design completion.

END OF SECTION 2

SECTION 3: Questions 27, 28, 29, 30, 31 and 50 deal with the basic approach to programming and design. Please: (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

50. The Architect's Guide to Facility Programming lists three basic approaches to programming, in your opinion, which of the following approaches is best [best describes your programming method].

(16) (52%) A. SEGREGATED: Programming is separate, distinct activity (1) performed prior to initiating of designing, and (2) performed by a separate individuals or teams from the designers.

(4) (13%) B. INTEGRATED: Programming is not a "redesign" service, but an integral first part of the design process.

(11) (35%) C. INTERACTIVE: Programming and designing are performed in alternating sequence and in response to each other.

A.	B.	C.	D.	E.
(5)	(4)	(3)	(2)	(1)
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

27. Programming should be integrated with conceptual design.

(3)	(8)	(7)	(13)	(0)
A.	B.	C.	D.	E.
(10%)	(26%)	(23%)	(42%)	(0%)

N = 31 MEAN = 3.032 MEDIAN = 3.000

28. Programming and conceptual design should be interactive, not separate phases of the facility delivery process.

(4)	(14)	(5)	(8)	(0)
A.	B.	C.	D.	E.
(13%)	(45%)	(16%)	(26%)	(0%)

N = 31 MEAN = 3.452 MEDIAN = 4.000

29. The programming process is the same for all facility projects.

(3)	(5)	(0)	(18)	(5)
A.	B.	C.	D.	E.
(10%)	(16%)	(0%)	(58%)	(16%)

N = 31 MEAN = 2.452 MEDIAN = 2.000

6

30. Programming is essential regardless of project size.

(7)	(15)	(1)	(5)	(3)
A.	B.	C.	D.	E.
(23%)	(48%)	(3%)	(16%)	(10%)

N = 31 MEAN = 3.581 MEDIAN = 4.000

31. The end product of programming is information not design.

(9)	(15)	(3)	(4)	(0)
A.	B.	C.	D.	E.
(29%)	(48%)	(10%)	(14%)	(0%)

N = 31 MEAN = 3.935 MEDIAN = 4.000

Definitions:

Conceptual Design - means conceptual or Schematic Design per A.I.A standard terms.

Design - means Design Development and Contract Document production per A.I.A. standards.

Comments:

Question 27

- [Programming should be integrated with design to] update programming as design proceeds.
- [Programming should be integrated with design] but only on firm projects.
- Again, it can be done [programming integrated with design]. What is wasteful is for a programmer to do a half-ass job. Then, design funds and time are wasted revising the concept.
- Sometimes it [programming being integrated with design] cannot be avoided.
- The regs [regulations] say it should be that way [programming integrated with design], but actually it is often more practical to wait until 50-75% design complete - and we do.
- [Programming should be integrated with design] for MILCON - 10% RAMP only. We cannot waste design effort on projects which will not be approved or funded up to 10% design.

Question 28

- [Programming and design should be interactive] but time between programming and design is often years.
- It could be done this way [programming and design being interactive].

Question 29

- The [programming] steps are [the same], the [programming] details are not. Money level often dictates depth of the documents.
- [The programming process] probably should be [the same for all facility projects] but it is not.

Question 30

- For the most part [programming is essential regardless of project size]. Exceptions will arise.

Question 31

- Normally yes [programming is essential], but there are exceptions.

END OF SECTION 3

SECTION 4: Questions 7, 8, 22, 23, 33, and 34 deal with the role of the programming document. (1) READ through all the questions and comments before answering, (2) SELECT the best answer, (3) CIRCLE your answer.

A.	B.	C.	D.	E.
(5)	(4)	(3)	(2)	(1)
STRONGLY AGREE	AGREE	UNDECIDED	DISAGREE	STRONGLY DISAGREE

7. Facility programming identifies the technical building requirements for design.

(3)	(8)	(2)	(14)	(2)
A.	B.	C.	D.	E.
(10%)	(28%)	(7%)	(48%)	(7%)

N = 29 MEAN = 2.862 MEDIAN = 2.000

8. A facility programming document is a problem definition or statement.

(4)	(18)	(1)	(5)	(2)
A.	B.	C.	D.	E.
(13%)	(60%)	(3%)	(17%)	(7%)

N = 30 MEAN = 3.567 MEDIAN = 4.000

22. A facility programming document is primarily information for the designer.

(4)	(18)	(3)	(9)	(4)
A.	B.	C.	D.	E.
(13%)	(35%)	(10%)	(29%)	(13%)

N = 31 MEAN = 3.065 MEDIAN = 3.000

23. A facility programming document is [primarily] valuable information for the user/using agency.

(0)	(1)	(3)	(25)	(1)
A.	B.	C.	D.	E.
(0%)	(3%)	(10%)	(83%)	(3%)

N = 30 MEAN = 2.133 MEDIAN = 2.000

33. A facility programming document should include the qualitative requirements of the user/using agency's organization.

(5)	(15)	(6)	(5)	(0)
A.	B.	C.	D.	E.
(16%)	(48%)	(19%)	(16%)	(0%)

N = 31 MEAN = 3.645 MEDIAN = 4.000

34. Programming should always produce a formal document.

(2)	(16)	(2)	(8)	(3)
A.	B.	C.	D.	E.
(6%)	(52%)	(6%)	(26%)	(10%)

N = 31 MEAN = 3.194 MEDIAN = 4.000

<u>Definitions:</u>

requirements - Refers to something wanted or needed, or a condition. Does NOT necessarily mean the extended treatment of or attention to particular items.

facility programming document - DD Form 1391, Project Book, RAMP or other documents that either give direction to designer, component prescriptions, design goals, alternative solutions or performance criteria.

qualitative requirements - Refers to requirements effecting the quality of the facility (i.e. organizational/personnel adjacencies, or work and traffic flow needs.)

<u>Comments:</u>

<u>Question 7</u>

- Agree [that programming identifies technical building requirements] to the point that the programmer doesn't design the job, but is able to identify technical items of significant cost impact.
- [Yes, programming identifies technical requirements] but not as detailed as the designer will get into.
- [programming identifies] unique technical requirements.

<u>Questions 22 and 23</u>

- [A facility programming document is information for designer] but also for justification and budgeting.
- [A facility programming document is] primarily an approval document, but almost equally information and primary direction for the designer.
- [A facility programming document is information for the designer] if done properly.
- No [a facility programming document is not primarily information for the designer]. It is to identify requirements/problems to approval authorities and for obtaining funds. By your definition - yes. A RAMP or project book is the document provide to designer.
- It [a facility programming document] helps user identify his requirements.
- Agree [that a facility programming document is primarily for designer], unless you're referring to a DD Form 1391.

Question 33

- [A facility programming document should include qualitative requirements] if needed.
- Sometimes [a facility programming document should include qualitative requirements.]
- Tend to think not [that a facility programming document should include qualitative requirements] but ok. This is a design related task more than programming. But if know during programming then it's ok to include it.

Question 34

- Nothing should be always.
- Neat hand lettering can be considered formal, too.
- Good but totally necessary [programming should always produce a formal document]. Definitely if design will be by AE. If in house probably not totally necessary.

END OF SECTION 4

SECTION 5: Questions 52, 53, and 54 ask which programming techniques are most effective for collecting, analyzing, organizing, communicating and evaluating data. Only techniques in which 40% or more of respondents marked as "have used" were included in the Round 2 questionnaire.

52. Which of the following techniques [have you used] are most effective when COLLECTING programming information.

(23)	(79%)	___	Background Data Research
(20)	(69%)	___	Surveys
(28)	(97%)	___	Interviews
(21)	(72%)	___	Direct Observation
(13)	(55%)	___	Participant Observation
		___	Others, Please Specify Below

53. Which of the following techniques [have you used] are most effective for ANALYZING and ORGANIZING programming data?

(11)	(41%)	___	Space Unit Standards
(11)	(41%)	___	Energy Budgeting
(26)	(96%)	___	Project Cost Estimating
(21)	(78%)	___	Construction Cost Estimating
(23)	(85%)	___	Life Cycle Cost Analysis
(18)	(67%)	___	Bubble Diagram
(15)	(56%)	___	Functional Relationship Diagram
(15)	(56%)	___	Layout Diagram
(13)	(48%)	___	Flow Diagram
(12)	(44%)	___	Organizational Chart
(13)	(48%)	___	Worksheets
		___	Others, Please Specify Below

54. Which of the following techniques [have you used] are most effective for COMMUNICATING and EVALUATING programming data?

(25)	(86%)	___	Brainstorming
(13)	(45%)	___	Buzz/Rap Session
(20)	(69%)	___	Group Planning
(24)	(83%)	___	Narrative
(23)	(79%)	___	Graphics
(13)	(45%)	___	Audio/Visual Aids
(24)	(83%)	___	Oral Presentations
(14)	(48%)	___	Panel Discussions
		___	Others, Please Specify Below

END OF SECTION 5

THANK YOU FOR YOUR PARTICIPATION.

Appendix D: Comments From Round One/Group A Questionnaire

The following are Group A's transcribed written comments and responses from the round one Delphi questionnaire (Appendix B). The comments are organized under the question to which they were responding. Comments not in response to a particular question are found at the end of the appendix, under general comments. The number before each comment is the respondent number, and its only significance is to the researcher for his own records. In addition, bracketed text is added by the researcher, in some cases, to clarify the context of the response.

6. Facility programming identifies the functional building requirements for design.

(13) & space (room by room).
(24) Absolutely.

7. Facility programming identifies the technical building requirements for design.

(13) [Also] furnishings, medical equipment (x-ray, ect.), engineering requirements.
(15) Not usually, although technical criteria and programming may be done together.
(18) We do, everyone doesn't.
(19) Usually, but may be a translation function of the design team.
(21) In some cases.
(24) In practice this is a follow up activity in certain design phases.

8. A facility programming document is a problem definition or statement.

(13) Often owners/users have to figure out their operational concept or hoe they are going to actually function before they can finalize the program. Most often the so called 'Final Program' undergoes adjustments as the user begins to see schematic design plans. More complicated relationships often generate more corridors, thus increasing the net to gross factor. Alignment or stacking of bldg. components too affect the program.
(18) I don't agree with (or think in) these terms.
(24) It [programming document] should state the clients goals, objectives and constraints.

9. A facility design is a problem solution.

(12) It [design] is more part of the definition of the problem.
(18) I don't agree (or think) in these terms.
(24) [The facility design] only should define the problem.

10. Programming is the responsibility of the client/owner.

(1) Not alone - shared responsibility.
(7) Program is client responsibility, but programming is architect's responsibility.
(8) Yes [client responsibility], if client can do it. Depend on definition [of responsibility].
(9) Programmer assists client/owner in making decisions which is their responsibility.
(13) [Yes,] though may be done by others.
(16) But most owners are not equipped to do [programming] so. A programmer facilitates and informs. Owners make the decisions.
(17) In reality, it [programming] becomes the responsibility of the owner. It should be the responsibility of the designer.
(19) Functional [programming is the responsibility of the client/owner].
(21) Most owners cannot develop a complete one [program].

11. Programming is the responsibility of the designer.

(1) [Programming is] team effort with client input; shared responsibility.
(8) Yes [designer responsibility], if retained to do it [programming].
(16) This is not in conflict with #10 [Question 10]. It is a joint effort. An architect - analytically based - must guide the process. A pure "designer" does not program well - "analysis vs. synthesis' mind-set.
(18) Depends.
(19) Design or architectural (form of AIA, B131) [programming is the responsibility of the designer]. Architect confirms the requirements of project to the owner as the first step in basic services.
(21) [Programming is] responsibility of the programmer or designer.
(24) The designer should participate or interface with the programmer for a complete project.

12. Programming is an iterative process.

(7) Yes, [when doing] the programming process only, same for design, but not iterative between program and design (not prog. design prog. design).

(13) Normally.
(15) [Yes], although re-programming may be needed if client's needs change.
(20) Yes, but the more you ask the same question, but in different ways, the more confused some clients get.
(24) Multiple client/user reviews are needed to develop consensus.

13. Design is an iterative process.

(7) [See comment for Question 12]
(20) Yes, but too many solutions confuse the client.
(24) Similar to [question 12]. Concept design should narrow the range of variations to make later design phases efficient.

14. Programming is a series of client design decisions.

(3) Decisions yes; design decisions, no.
(13) [Programming is] an extensive [series of client design decisions].
(16) [Programming is series of] informal [client design decisions]. "Design" being design guidance, not solution.
(17) Client input is important; there are no decisions, there are guidelines.
(18) [Programming is series of client] design-related [decisions].
(24) Programming sets direction/concepts for design and eliminates options.

15. Client/user participation is very important in programming.

(13) User should include key dept mgrs and technicians, CEO, housekeeping, security, ect.
(24) A must! Even in highly centralized, top-down organ. the end users will influence program and design.

16. A programmer should guide clients through decision making.

(13) Programmers will quickly find themselves in the middle of internal disputes. If operational decisions are not made yet by user, then programmer should request owner/user CEO or project manager to secure answers for the next meeting. Much of 'programming' meetings can be wasted while owners users debate. Programmer should raise the operational questions and options and make cleat the decisions required to make the program.
(19) Help, not guide.
This seems to imply that programming is a consultant to owner, but more and more organizations have programmer as an internal consultant, in the facilities group.

(24) [A programmers should guide clients] to the degree possible choices should be discussed.

17. Clients/users should be part of the programming team.

(19) Programming should be an iterative process, so team work becomes inevitable.
Programming is giving leadership.
(24) Same [as Question 15].

18. Designers should be part of the programming team.

(7) [Designers should not be part of the programming team] unless they can be objective.
(16) "Designer" no - architect (analytically inclined) yes.
(18) Depends, [designers as part of the programming team] can work very well (or not).
(19) [Designers should be part of the programming team] as observers and contributors.
(20) A clear program should allow designers to proceed.
(24) [Designers as part of the programming team], a must for effective follow through - the designer always finishes the program in practice.

19.It is important to educate the client/users in the programming process.

(18) Depends on the client and their experience.
(19) What are you assuming about client/users? Usually they are not the same. users = occupants, client = facilities group or top executives.
(24) It helps communication and avoids misunderstanding and excessive expectations.

20. It is important to educate the client/users in the programming process.

(8) [The] point is to lead client thru the process of design.
(13) [It is important to educate the client/users in architectural design] process and levels of information development.
(16) In their understanding of the foundation of function, scale, context.
(18) [Educating clients/users in design] depends on the client and their experience. Also, [design] is not really necessarily part of the programming phase.
(20) Most clients need to understand design approach or design direction to give meaningful data to the programmer.

21. Three-way communication between the designer, programmer, and client is essential to programming.

(7) Communication must establish a common language, but this does not mean designer communication with client during programming.
(8) Designer and programmer - these can be one person or two. A really good programmer can produce a document without the designer.
(18) [Objected to "essential" - wrote can be "valuable"]
(21) Four-way communication - include users.
(24) Absolutely.

22. A facility programming document is primarily information for the designer.

(1) But a sophisticated client will use the programming document as management tool and for future requirements database.
(3) It's [programming document] both for the designer and client.
(8) It [programming document] has important uses for client.
(12) It [programming document] also conveys, verifies, or questions owner/user perceptions.
(16) Equally for owner and designer - it serves as guide and framework for both.
(17) This word [primarily] leaves me undecided, if it were out I would have checked A [strongly agree].
(18) Though also for client.
(19) [Circled "D" - disagree, circled "A" - strongly agree for designers] and users, [circled "B" - agree] for facilities professionals within the client organization.
(21) Depends on project assignment.
(23) [Programming document] both [for] designer and client. Client needs to have their expectations articulated so they can participate knowledgeably in the design decision process.
(24) Also for operational planning and financial planning.

23. A facility programming document is primarily information for the client.

(1) Also a critical document for design of the facility.
(3) [See comment for Question 22].
(7) Document is feedback to client for approval.
(8) [It] is equally important to designer.
(13) It [programming document] is often a commitment document of the staff and CEO, if it is used as a sign-off document. In the case of an A/E [Architect/Engineer] contract, the program is a contract document on which fees and project costs are based.

(15) [Programming document is] primarily for designer but also a valuable reference for client.
(16) [See comment for Question 22].
(17) [See comment for Question 22].
(18) [See comment for Question 22].
(19) [Circled "B" - agree for] clients, [circled "A" - strongly agree for] occupants. [Occupants] will get as much as designers.
(20) [A programming document] states client's needs, scope.
(21) [See comment for Question 22].
(24) The client is responsible for approving the program. The document is his contract with the designer.

24. Conceptual design and contract documents production are two separate phases of the design process.

(1) But [we should be] keeping construction details in mind when doing concepts.
(8) It's a flow of document development - one process
(12) Except in small projects - where there may be one continuous "phase".
(13) The military should use phases and terminology common to the industry. All architects are trained and practice to the A.I.A systems. Schools are accredited by using the A.I.A system. Much time and energy is wasted both by gov't employees and civilian contractors making the conversions. A.I.A phases include: [1] programming phase, [2] schematic design phase, [3] design development phase, [4] contract documents phase, [5] bid/negotiation phase, [6] construction phase, [7] post-occupancy evaluation.
(17) As practiced, it [conceptual design and contract documents production] is iterative.

25. Conceptual design is part of the programming process.

(1) Preferably [conceptual design is part of the programming process], but sometimes is separated successfully.
(9) What is this? [referring to conceptual design]
(12) Otherwise there is no way of testing the value of [the programming].
(13) [Conceptual design] adjusts the program.
(17) It [programming process] is iterative.
(18) [Conceptual design is not part of the programming process] but [it] can be valuable to explore design implications.
(21) Depends on the client and schedule.
(23) I am now engaged in trying to integrate programming and design in several projects, as an experiment. So far, it seems to make sense, but I am still undecided on these matters.

(24) They [conceptual design and programming] can be mutually supportive and time saving to do coordination with schematics, except that detailed technical info (data sheets) can wait.

26. Programming should be completed prior to design.

(1) [Yes programming should be completed prior to design] except to evolve into concepts as noted in #25 [Question 25] above.
(8) Yes [programming should be completed prior to design], but not always possible or desirable; design often tests program assumptions or requirements.
(13) A [programming] document [should be completed prior to design.
(15) Answer assume[s] that "design" means contract documents per your note in 47(A) [Question 47]. If design were as customarily defined (i.e. schematic design, design development) I would answer these questions differently.
(16) Various levels of programming exist - each phase has a "programming" element to it.
(18) Depends; we now (sometimes) do schematic level program [first]; then [complete] detailed program while design is underway. [There] are, however, risks (that change [requirements] as get into detail).
(20) Client must determine scope [of programming].
(21) Depends on client and schedule [if programming is completed prior to design].
(23) [See comment for Question 25].

27. Programming should be integrated with design.

(1) [Programming] must be [integrated] early in design process. As noted in 25 and 26 [Questions 25 and 26] above.
(7) Subject (content) should be integrated not the process.
(15) [See comment for Question 26].
(17) No [programming should not be integrated with design], it is an iterative process with design.
(18) I'd say [programming should be] coordinated [with design not integrated].
(19) [Programming should] continue concurrently [with design]. As originally worded [circled "E" - strongly disagree].
(21) [Programming should be] integrated with conceptual design, but some programming is required to do it.
(23) [See comment for Question 25].
(24) Only concepts [should be integrated with design].

28. Programming and design should be interactive, not separate phases of the facility delivery process.

(13) Trial designs should be tested during the programming process to verify interrelationships, and net to gross factors.
(15) [See comment for Question 26].
(16) As long as it's structured.
(18) Yes [to interactive, but] no don't agree generally [to separate phases].
(20) We put both in Phase I services. See [comment for] Question 47.
(21) Base building program should proceed schematic and design development.
(23) [See comment for Question 25].
(24) [Programming is interactive] with concepts.

29. The programming process is the same for all facility projects.

(1) [Programming process is] similar, not the same.
(8) Nothing is the same [programming process] for all projects.
(20) [The programming process is] never [the same]; format similar, technique[s] not standard.
(24) Variations in client terms, data availability, schedule milestones (financial, marketing, PR, ect.) can all influence [the programming] process for each project.

30. Programming is essential regardless of project size.

(12) At some level.
(23) I don't think size is the issue. I think familiarity with the type of facility is the issue. If the project is one that is a well-known type, for a 'typical client', perhaps programming is not required.

31. The end product of programming is information not design.

(1) If not conceptual design, a certain amount of design guidelines and design criteria must be included in programming.
(9) [See comment for Question 25].
(12) The end product of it all is a project that fulfills the requirements set for it.
(13) Many programs really drive design.
(19) [Circled "E" - strongly disagree] as originally worded. [Would strongly agree if "information" was replaced by] decision making.
(20) We say that the program is the roadmap to design.
(21) Depends on the project [if programming is information not design].

(24) [The end product of programming is information not design] except that concept design does: (a) confirm program concepts and (b) sets firm design directions.

32. A facility programming document should include the quantitative requirements of the client's organization.

(12) Where they pertain.
(24) Everything available and positive

33. A facility programming document should include the qualitative requirements of the client's organization.

NO COMMENTS

34. Programming should always produce a formal document.

(1) Preferable - but may not be practical within a given budget.
(12) That document may be the design.
(19) Usually, but not always.
(21) Depends on schedule and process.

35. A programmer should have experience in design.

(1) [A programmer should be] sensitive to design but not mandatory to have experience in design.
(3) [A programmer should have experience in design] at least in school.
(8) [A programmer with design experience is] desirable, but not critical.
(12) Or work closely and well with people who do.
(15) Design experience certainly helpful, but I have trained people without design experience to be excellent programmers.
(16) [Programmer] should have a strong understanding in architecture and planning; but not a pure "designer".
(18) Depends; someone on [programming] team should [have experience in design.
(19) [A programmer with design experience is] helpful but not essential.
(24) It [a programmer with design experience] is very helpful but not essential.

36. A programmer should be competent in communication skills, including graphic analysis and display.

(3) But primarily written and verbal.
(8) Graphic analysis and display - nice, but.
(9) Why not.

37. A programmer should understand the whole building delivery process.

(3) [A programmer should understand the building delivery process] at some level.
(8) [A programmer should understand the building delivery process] - not necessarily.
(13) This [understanding the building delivery process] certainly helps. I've seen many owner[s] who have retained prominent number crunching firms or planning-only firms who have produced incredibly poor and understated programs and budgets, including the military. Then a knowledgeable A/E is brought on board, and he has to begin a business and personal relationship by telling the owner a lot of bad news, or risks being liable for the eventual consequences. Then the budget is upset while programming is revised and additional funding secured, while time flies by and escalations erodes the budget further.
(16) [Undestanding] can be somewhat lighter in the construction phase.
(18) Someone on [programming] team should [understand the building delivery process].
(24) A general understanding [of the building delivery process] is needed.

38. During the programming process, uncovering the true needs of the client is a recurring problem.

(3) True needs? - bad question.
(18) Issue? challenge? = yes; problem = no.
(21) Its a challenge, not necessarily a problem
(24) The programmer responds to the problem as defined by the client - except to the degree the programmer personal experience and stature with the client influence the client perceptions.

39. During the programming process, getting clients to make decisions is a recurring problem.

(8) Just depends on the client - also on how you ask and manage process.
(17) Varies from client to client.
(18) [See comment for Question 38].
(21) [See comment for Question 38].
(24) Seldom will they [the client] take responsibility for their input and directions.

40. During the facility delivery process, the programming - design relationship/connection is a recurring problem.

(1) [The programming/design connection] should not be a problem; integrated team is best.

(16) Seems to be for most firms. Architects generally do not comprehend "programming" as a discipline.

(19) Problem [referring to programming/design connection] is architects and other professionals who have been educated to believe that they know best! and do not know how to listen.

(23) [Programming/design connection is a problem] only if the client abandons the program as the formal embodiment of their needs, or if was a lousy program to begin with.

(24) [If the programming/design connection is a problem] depends on: (a) designer involvement in program, (b) implementation of the program in its detail, (c) stability of the program related to client changes.

41. How many opportunities, on the average, do your clients have to review, verify, change or add to the programming information?

(1) 1 - Initial Input; 2 - First Draft Review; 3 - Additional Input; 4 - Final Program Review.

(7) During what period? [Answered 5 or more] during programming.

(17) Varies from project to project.

(18) [Answered 4] Sometimes more if needed. This is of documented products.

42. How many design solutions, on the average, do you or your firm present the client/owner?

(1) [Answered 3 as] average. Sometimes less, sometimes more.

(8) [Answered 3] but very often only one. When you find the right approach, no point in wasting time and money.

(17) Varies from project to project.

(18) Not applicable.

(19) Not applicable.

(24) [Answered 3] as a goal.

43.In your opinion, what percentage of overall project development time should be spent on programming.

(3) What is this [referring to project development]. Absolutely dependent on the project specifics - size, goals, complexity, etc.

(8) Impossible to answer. Complex programs can result in simple, quick solutions and [the] reverse is also true.

(13) Really varies. More time, money and expertise retained early in the programming phase can save time and money in the design phases.

(18) % [percentage] depends on project. Programming for moderately complex project [can] take 3 - 6 months.

(19) Concurrent through to programming the furniture layouts before move in.

44. You or your firm use a computer (not including word processing) to perform _________ of the analyzing, organizing, and evaluating of programming data.

A. MOST
B. SOME
C. LITTLE
D. NONE

(18) We do all diagrams and space lists on computer.

45. Programming is:

A. PART OF THE DESIGN PROCESS
B. SEPARATE FROM THE DESIGN PROCESS

(3) Both.
(8) It's [programming as part of the design process] best, but can be done separately very successfully - but it's much harder.
(12) It is both. It has its own discipline and should be done concurrently and interactively with design.
(15) Answer assume[s] that "design" means contract documents per your note in 47(A) [Question 47]. If design were as customarily defined (i.e. schematic design development) I would answer these questions differently.
(17) Neither - It is interactive with the design process.
(18) As practiced by us [programming is separate from the design process].
(19) [Programming is] interactive with the design process.

46. Conceptual design is:

A. PART OF THE PROGRAMMING PROCESS
B. PART OF THE DESIGN PROCESS
C. PART OF BOTH THE DESIGN AND PROGRAMMING PROCESSES
D. SEPARATE FROM THE PROGRAMMING AND DESIGN PROCESSES

(8) [Conceptual design is part of the design process] but best shared during programming.
(9) What is this? [referring to conceptual design]
(15) [See comment for Question 45]
(20) [Conceptual design is] include in Phase I. [See comment on Question 47]

47. The distinct phases of the facility delivery process are:

A. PROGRAMMING, CONCEPTUAL DESIGN, DESIGN, and CONSTRUCTION.
B. PROGRAMMING, DESIGN, and CONSTRUCTION
C. DESIGN and CONSTRUCTION
D. OTHER

(8) B is the same answer [as A] - it's all part of process.

(13) [Use terms] schematic design, design development [referring to conceptual design and design]. Use A.I.A standards.

(16) Programming; Conceptual; Design Development; Construction Documents; Bid/Award; Construction.

(17) Programming, Conceptual Design and contract documents are iterative.

(18) Strategic Planning (needs assessment/project identification); Programming; Design; Construction; Activation; Evaluation; Retirement.

(19) Programming is a concurrent activity which starts ahead of the design process and continues until after move-in.

(22) I use an iterative approach with overlapping programming phases, Our design phases are concept, schematic, design development, construction documentation.

(24) [Programming, Conceptual Design are] jt [joint] activities. [Design] should be schematic, design development, contract documents.

48. In your opinion, who should control the programming of facility projects.

A. CLIENT/OWNER
B. DESIGNER OR DESIGN TEAM
C. IN-HOUSE PROGRAMMING STAFF (part of the design firm)
D. OUTSIDE PROGRAMMING CONSULTANTS (separate from the design firm)

(8) Assumes client has no in house capability.

(10) Varies by project conditions and project type and designer/owner expertise.

(12) Taking "control" literally, the client should control all aspects of the project.

(13) [Answers] B, C, D should [all] have strong influence.

(17) Small firms cannot always have in house staff - they might require consultation.

(18) We are this category [in-house programming staff], but client controls. Can't generalize [on the question since answer] varies by type/competence of clients.

(19) Programmer should be either on staff of client/owner or a direct consultant to them. Client controls.

(22) C [In-house programming staff] is best but only if qualified programming professionals are on the design staff, which is rare. If not, D [outside programming consultants] is best. I other words, quality is most important, then integration w/design team; preferably both are provided.

(24) Client always has responsibility for decision, design team should usually be responsible for the programming process."

49. **Programming includes:**

A. **DETAILS FOR CONTRACT DOCUMENTS PRODUCTION**
B. **MAJOR ISSUES FOR CONCEPTUAL DESIGN**
C. **BOTH**

(1) [Answered "C"] but mainly "B".
(8) [Answered "C"] and that is the reason the design team should control - to assure complete and meaningful information.
(12) [Answered "C"] with emphasis on "B".
(18) [Answered "C"] we do detailed requirements - not drawn design details (confusing).
(20) Never "A".
(21) [Answered "B"] "C" is true on certain projects.

50. **The Architect's Guide to Facility Programming lists three basic approaches to programming, which of the following approaches best describes your programming method.**

A. **SEGREGATED**
B. **INTEGRATED**
C. **INTERACTIVE**

(8) [Answered integrated] but include C [interactive] in answer. Programming is the first step in the process, but it often becomes interactive as it is tested during early design stages.
(13) Normally, when a programming phase is completed w/o any design input, it is made clear that some adjustment will be made in the schematic design phase.
(17) [Answered both segregated and interactive] Depends on the job.
(21) A [segregated] and C [interactive] depending on project.

51. **A programming document almost always should include:**

(1) History
Trends
Personnel Projections
[Budget and Cost Information] usually separate but concurrent.
(2) Near, Mid, Long Range staff and space needs.
Code Implications
Parking
(7) Prospects for change and growth.
(8) Whatever else is known.
(9) Area or Sq. Ft. of facility.
"Image" or appearance issues.
(12) Investment criteria for making cost-related decisions.
Owner's decision power.

(13) Functional Relationships
Bubble Diagrams
Operational Concepts
Circulation Patterns
(15) Net area breakdowns.
Estimated gross area.
(16) Priorities
Adjacencies
(17) Symbolic and Aesthetic Criteria
[Technical Requirements] occasionally
[Schedule Information] occasionally
(18) [Budget and Cost Information; Schedule Information; Energy Requirements] optional - depends.
(20) Noise, data, video needs.
Interior Air Quality Standards
[Referring to Energy Requirements] Most designers will follow code standards, rarely quality of light can be determined.
(22) Operational [and] Human Resources
Behavioral Goals and Criteria
(23) Descriptions of mission and activities of each subgroup of the organization and for each functional job category (defined as people exhibiting common task behaviors, regardless of task content)
Appropriate images for the place, based on the organization's culture.
(24) We see these [Technical Requirements; Environmental Data; Energy Requirements] as follow on activities in the design process.

52. Which of the following techniques have you used when COLLECTING programming information.

(6) Document Analysis
Archival Records
(8) I don't know what these mean. [referring to 6 of the possible answers] Sounds like someone's buzz words. Programming is not that difficult - it takes knowledge of buildings and some brains to ask the right questions. Very often it's overworked. Perhaps to justify fees. Large, bound professional looking programs are often 80% of the shelf B.S..
(13) Extensive and intensive user interviews to ferret out the real needs.
Visiting other facilities with user in tow.
(18) Facilitated Group Discussion/Decision Process
Facility Tours
Touring Interviews
Photographic Documentation
Literature Reviews
POE's of related projects
(19) Photo Survey
Time-Lapse Survey

(20) Relationship Diagrams

Most of our work is technical therefore data gathering is very critical. Long range planning (capacity planning) is a combination of client history, our experience, industry trends, current technology, future technology, and very flexible space.

(25) Focus Groups

53. Which of the following techniques have you used when ANALYZING and ORGANIZING programming information.

(7) Site Analysis Diagrams
Climate Analysis Chart

(8) [See comment for Question 52]

I have used highly sophisticated math techniques once in over 30 years. That was for a long range facility needs forecast for a state capitol and entire state's facility needs.

(13) Trial net to gross factors to use for different functions in renovation work. This is very important. In health care projects, some functions lay out very inefficiently in existing space configuration. If trial designs can not be obtained during programming, then a contingency should be allowed for this. It often takes substantial design effort to determine what functions will fit into existing space, and how well they fit.

(17) Statistical Modeling

(18) [Use but] try to avoid [Relationship Matrices].

(20) Keep it simple.

(23) I think feedback sessions are an important techniques for establishing some consonance between programmers' perceptions and interpretations and those of the client and users.

(25) Spreadsheets (net-to-gross, support space, circulation, ect.)

54. Which of the following techniques have you used when COMMUNICATING and EVALUATING programming information.

(6) Video, other media
Workshops

(8) Most of this is B.S. [referring the list of possible answers]

(13) Visiting other facilities with owner/user and evaluate those facilities.

(18) Nominal Group Technique

(20) Design by committee does not work. Charrette's work only if design team leader knows when to stop compromises. Design direction must be based on a concept design.

55. What two or three questions would you like to ask your peers about facility programming?

(1) 1. What techniques are most effective for obtaining client input?

(2) 1. Request examples of buildings which benefited from effective programming and conceptual planning.

2. Explain how programming has resulted in more appropriate projects.

3. How can programming assist in incorporating flexibility to meet future needs?

(3) 1. Any interest in a "Facility Programming Association"?

(6) 1. Is it desirable to standardize approaches to facility programming?

2. What are the professional liability implications of F. P.?

3. What fee schedule is appropriate for programming services?

(9) 1. How do they distinguish between programming and design?

2. What part of their firm's practice is programming?

3. How did they learn about programming?

(13) 1. Requirements for the procurement of qualified programming services either in-house or outside contracts.

(15) 1. Often a programmer is put in the middle of client-user conflicts or political in fighting. How do you deal with these situations?

2. There are many potentials for misunderstandings due to different forms used in defining space, e.g. net, usable, carpeted, gross, departmental gross, rentable, loss factor, ect.. How do you explain these terms to your client to ensure that client, landlord. developer, and/or all speak the same language?

(16) 1. Where can I find programmers?

2. What schools have enough programming courses to graduate reasonably competent programmers/analysts?

(17) 1. Fee structures?

2. How to make clients and architects more aware of the uses and benefits of facility programming?

3. Should there be professional licensing of facility programmers?

(18) 1. What set of techniques best helps ensure robust programming decisions (that last thru design and occupancy)?

(23) 1. Given that clients need a programming - design service, what are reasonable ways to organize to provide that service, to sell it as an addition (to an information impoverished design process) and how to develop and maintain standards of performance in the programming process.

(25) 1. What type of archival information is maintained in databases (i.e. net-to-gross ratios, office-to-support, area/person)?

2. Is space standards analysis a typical component of programming?

3. How is it determined who should be interviewed in a programming effort? Who approves the information?

General Comments

(8) You have a good thing going. But a lot of phrases are selling tool words from both programmers and designers. It's really a simple process. When kept simple and direct it easily interfaces with design processes. Programming is a simple tool to communicate client needs to the design team. good designers do it very well. They know the questions and how to find the key problems.

(15) Some of my colleagues take strong positions on the scope and nature of facility programming. E.G. Programming is design. Programming is a discrete activity or it encompasses many activities. Programming is not design. Programming should be done only by designers. Or never by designers.

I consider these distinctions less important than the fact that there are a number of interrelated tasks that need to be done before conceptual design begins, and that programming is a unifying element in these tasks. I have found it useful to call these services "predesign".

Every project is different. Not all predesign services may be necessary (although some form of programming is always needed). Sometimes it's expedient to include special studies, such as audio visual or space utilization, in a program document.

I'm enclosing a few diagrams to illustrate these points.

(20) Architects and engineers have not (are not) designing buildings any differently since the advent of air conditioning in the late 1940's. I encourage my clients to embrace a concept I call P.O.P Architecture (Point of Presence Origin). This theory challenges the client and the design team to prepare space to easily change from type I space to type II space quickly and simply, but more than anything else allow the clients to enjoy the benefits of the most contemporary office equipment, indoor air quality, and still be able to open the windows of their buildings. The trend of the 90's is to have highly technical space, with proper HVAC, power, light, telecom, video, ect.. Solve these issues at the Point of Presence of the end user. The challenge then is to allow dramatic change to occur without disturbing the infrastructure of the building.

Appendix E: Comments From Round One/Group B Questionnaire

The following are Group B's transcribed written comments and responses from the round one Delphi questionnaire (Appendix B). The comments are organized under the question to which they were responding. Comments not in response to a particular question are found at the end of the appendix, under general comments. The number before each comment is the respondent number, and its only significance is to the researcher for his own records. In addition, bracketed text is added by the researcher, in some cases, to clarify the context of the response.

6. Facility programming identifies the functional building requirements for design.

(30) Should (but people try to consider it as a final design).

(36) [Facility programming] not a DD 1391 [identifies the functional building requirements].

(38) [Strongly agree] based on your definition. [Disagree] based on classical programming. If you are considering MILCON only, a recent procedure is to prepare a "RAMP" which is similar to a project book. However this is the start of the design phase - not programming.

7. Facility programming identifies the technical requirements for design.

(12) Sometimes.

(27) Agree [that programming identifies technical building requirements] to the point that the programmer doesn't design the job, but is able to identify technical items of significant cost impact.

(30) [Yes, programming identifies technical requirements] but not as detailed as the designer will get into.

(38) [See comment at Question 6 above].

(39) [programming identifies] unique technical requirements.

8. A facility programming document is a problem definition or statement.

(38) AF Form 332 or DD 1391 states requirement and current situation and is used as approval document.

9. A facility design is a problem solution.

(12) To some degree.
(27) However, the programmer must have a good idea of the probable solution to have his cost estimate within 25% of the final CWE.

10. Programming is the responsibility of the user/using agency.

(27) But if they [the user] don't get serious at the programming stage, much time and money is wasted later in the process.
(30) Knowing what they [the user] need and when - yes; documents - no.
(36) Designer should solve the users' problem, not just design what the user wants or think he wants.
(38) User identifies requirement/problem - CE is responsible for programming using user input.

11. Programming is the responsibility of the designer.

(27) However, the programmer should use his technical staff to pindown the scope.
(30) Only to the point of keeping the programmer informed of changes.
(33) Designer must ensure project is programmed correctly.

12. Programming is an iterative process.

(27) What do you mean?
(30) [No, programming is not an iterative process] unless you are referring to 'how' to fill out the paperwork.

13. Design is an iterative process.

(27) What do you mean?
(38) Hopefully [design is] not [an iterative process] for a given problem design. As a process yes.

14. Programming is a series of user/using agency design decisions.

(27) Actually, we [Civil Engineering] make the decisions. The user defines the problem.
(36) User should not be making design decisions.
(38) [Programming is] identifying user requirements and criteria.

15. User/using agency participation is very important in programming.

(38) Essential.

16. A programmer should guide users/using agencies through decision making.

(38) And inform (provide understanding or process).

17. Users/using agencies should be part of the programming team.

NO COMMENTS.

18. Designers should be part of the programming team.

(23) But designers are not available to do this [be part of programming team.]
(38) [Designers should be part of the programming team] for information and input. However some designers become worried over funding and block design decisions.

19. It is important to educate the users/using agencies in the programming process.

(30) Especially the time it takes for conception to completion.
(34) It isn't often done. And users aren't often willing participants.

20. It is important to educate the users/using agencies in architectural design.

(27) [Users should be educated in architectural design] only if they insist on defining a solution which is architecturally incorrect.
(38) [Users should be educated in architectural design to] provide an understanding and as part of the team.
(39) They [users] need to be aware of architectural compatibility.

21. Three-way communication between the designer, programmer and user/using agency is essential to programming.

(23) Again lack of designer time is a problem here.
(36) The programmer is the designer under your definition in the general instructions.

22. A facility programming document is primarily information for the designer.

(23) [A facility programming document is information for designer] but also for justification and budgeting.
(27) [A facility programming document is] primarily an approval document, but almost equally information and primary direction for the designer.
(30) [A facility programming document is information for the designer] if done properly.
(36) Agree, unless you're referring to a DD Form 1391.
(38) No [a facility programming document is not primarily information for the designer]. It is to identify requirements/problems to approval authorities and for obtaining funds. By your definition - yes. A RAMP or project book is the document provide to designer.

23. A facility programming document is primarily information for the user/using agency.

(36) [See comment at Question 22].
(38) It [a facility programming document] helps user identify his requirements.

24. Conceptual design and contract documents production are two separate phases of the design process.

(2) One feeds into the other.

25. Conceptual design is part of the programming process.

(23) [Conceptual design] may be desirable to be [part of the programming process] but not possible with current manning.
(27) Is done both ways [conceptual design either as part or separate from programming process.] The better the programming document, the fewer surprises in the concept design.
(34) Sometimes [conceptual design is part of the programming process], but not often.
(38) [Conceptual design is part of the programming process] for MILCON program. [Conceptual design is separate from programming process] for O&M program.

26. Programming should be completed prior to design.

(11) Initial approval and scope should be planned in advance to allow design scheduling.
(23) If not there is no approval. Can't spare design time on wishes.
(30) If time permits.
(38) This is the way MILCON works.

27. Programming should be integrated with design.

(11) [Programming should be integrated with design to] update programming as design proceeds.
(23) [Programming should be integrated with design] but only on firm projects.
(27) Again. it can be done [programming integrated with design]. What is wasteful is for a programmer to do a half-ass job. Then, design funds and time are wasted revising the concept.
(30) Sometimes it [programming being integrated with design] cannot be avoided.
(34) The regs [regulations] say it should be that way [programming integrated with design], but actually it is often more practical to wait until 50-75% design complete - and we do.
(38) [Programming should be integrated with design] for MILCON - 10% RAMP only. We cannot waste design effort on projects which will not be approved or funded up to 10% design.

28. Programming and design should be interactive, not separate phases of the facility delivery process.

(23) [Programming and design should be interactive] but time between programming and design is often years.
(27) It could be done this way [programming and design being interactive].
(34) Here it is.

29. The programming process is the same for all facility projects.

(30) The [programming] steps are [the same], the [programming] details are not. Money level often dictates depth of the documents.
(38) [The programming process] probably should be [the same for all facility projects] but it is not.

30. Programming is essential regardless of project size.

(11) [Programming and] planning [are essential regardless of project size].
(36) For the most part [programming is essential regardless of project size]. Exceptions will arise.

31. The end product of programming is information not design.

(34) Normally yes [programming is essential], but there are exceptions.

32. A facility programming document should include the quantitative requirements of the user/using agency's organization.

(27) If necessary.
(30) As much as possible.

33. A facility programming document should include the qualitative requirements of the user/using agency's organization.

(27) [A facility programming document should include qualitative requirements] if needed.
(34) Sometimes [a facility programming document should include qualitative requirements.]
(36) Tend to think not [that a facility programming document should include qualitative requirements] but ok. This is a design related task more than programming. But if know during programming then it's ok to include it.

34. Programming should always produce a formal document.

(23) Nothing should be always.
(30) Neat hand lettering can be considered formal, too.
(36) Good but totally necessary [programming should always produce a formal document]. Definitely if design will be by AE. If in house probably not totally necessary.

35. A programmer should have experience in design.

(23) Sure would help [experience in design.]
(27) [The programmer] needs to be able to understand his function vs. design.
(30) [Experience in design is] beneficial but not mandatory. Hard to find designers that are willing to be programmers.
(35) [A programmer needs experience in design but] doesn't need much though.
(36) [A programmer] should be the designer not a 1391 writer.
(38) It [experience in design] would be nice but not necessary.
(39) Ideally [a programmer should have experience in design.]

36. A programmer should be competent in communication skills, including graphic analysis and display.

NO COMMENTS.

37. A programmer should understand the whole building delivery process.

(2) What is it? [referring to whole building delivery process]

38.During the programming process, uncovering the true needs of the user/using agency is recurring problem.

(23) Changes with people in using agency.
(27) May be the biggest problem.

39.During the programming process, getting the users/using agencies to make decisions is a recurring problem.

(23) And sticking to their decision.
(30) To stick with a decision is even harder.

40. During the facility delivery process, the programming - design relationship is a recurring problem.

(30) Remembering to keep programmer informed of changes in concepts [is a problem.]
(38) Depends on program [if design/programming connection is a problem.] On some projects. Usually when new requirements are identified 9user, mission changes. regulations) or funding was approved prior to design completion.

41. How many opportunities, on the average, do your users/using agencies have to review, verify, change or add to the programming information?

(27) At least once in programming phase. About 3 times in design phase.

42. How many design solutions, on the average, do you or your A-E firm present the user/using agency?

(11) [Answered 2] mainly due to funding constraints.
(27) Sometimes one, sometimes 2 or 3. Depends on the problem.
(38) [Answered 3 for] MILCON 10% RAMP. [Answered 1 for] other programs.

43.In your opinion, what percentage of overall project development time should be spent on programming.

(38) There is no answer. It depends on the project. Size, complexity, cost.

44. You or your firm use a computer (not including word processing) to perform __________ of the analyzing, organizing and evaluating of programming data.

A. MOST
B. SOME
C. LITTLE
D. NONE

(36) Still developing our in house computer capabilities.

45. Programming is:

A. PART OF THE DESIGN PROCESS
B. SEPARATE FROM THE DESIGN PROCESS

(27) [Separate from the design process] at our base.
(38) [Programming is part of] project development. By your definition it is part of the design process.

46. Conceptual design is:

A. PART OF THE PROGRAMMING PROCESS
B. PART OF THE DESIGN PROCESS
C. PART OF BOTH THE DESIGN AND PROGRAMMING PROCESSES
D. SEPARATE FROM THE DESIGN AND PROGRAMMING PROCESSES
E. OTHER

(27) [Conceptual design is part of the design process] currently. Could be any way you want, as long as it is done.
(38) [Part of the programming process] for MILCON. [Separate from the design and programming processes for] O&M.

47. The distinct phases of the facility delivery process are:

A. PROGRAMMING. CONCEPTUAL DESIGN, DESIGN, and CONSTRUCTION
B. PROGRAMMING. DESIGN, and CONSTRUCTION
C. DESIGN and CONSTRUCTION
D. OTHER

(13) [Would add] environmental review [to answer "A"].
(27) Could be "A" or "B" depending on your desires. I'm not sure which is best. Except that we get poor programming documents with the system described in "A".

48. In your opinion, who should control the programming of facility projects.

A. USER/USING AGENCY
B. DESIGNER OR DESIGN TEAM
C. IN-HOUSE PROGRAMMING STAFF
D. OUTSIDE PROGRAMMING CONSULTANTS (A-E firms)
E. OTHER

(7) In-house staff with user input.
(11) Project management team [to include programming, design, and construction].
(15) User, programmer, designers.
(38) Our design team includes user, zones, designer, construction management, fire, safety, environmental, communications on all teams plus others as required.

49. Programming includes:

A. DETAILS FOR CONTRACT DOCUMENTS PRODUCTION
B. MAJOR ISSUES FOR CONCEPTUAL DESIGN
C. BOTH

NO COMMENTS.

50. The _Architect's Guide to Facility Programming_ lists three basic approaches to programming, which of the following approaches best describes your programming method.

A. SEGREGATED
B. INTEGRATED
C. INTERACTIVE

(11) [Answered segregated but] not necessarily the preferred way.

51. A programming document almost always should include:

(10) Constraints
(13) Furnishings
O&M Manuals
(23) Special [Technical Requirements]
[Schedule Information] if urgent.
(24) Impact Statement
(33) Justification (Need)
(38) Impact if not approved
Related work by other projects
Classifications of work
(39) Special design criteria [referring to technical requirements].
Need date or phasing requirements [referring to schedule information].

52. Which of the following techniques have you used when COLLECTING programming information?

(23) Base Comprehensive Plan

(35) Basicly, talk with using organization and find out what they want, then work with them to produce the desired product.

(38) For RAMP? Project Book? Or for programming?

53. Which of the following techniques have you used when ANALYZING and ORGANIZING programming information?

(33) As Builts (for renovations)

(35) See answer for #52 [Question 52].

54. Which of the following techniques have you used when COMMUNICATING and EVALUATING programming information?

(38) Base Facilities Working Group (all units represented)

Base Facilities Board (all senior staff from all units)

Monthly "How goes it" with all units represented to include Base Commander

55. Do you believe Air Force programming methods adequately define project requirements prior to initiating design? Please explain.

(4) Yes for base level projects. Problems occur on MCP and downward directed projects for MAJCOM.

(5) Yes. We will always have problems related to workload, costs and cost limitations, prioritizing and communications with required parties. Manpower changes frequently cause problems.

(7) No. We should move to a total integrated process [of] program, design, construct, evaluate.

(10) Reference to MCP projects. Absolutely not! Average time from need identification to completion is five years minimum; 6 - 8 is normal! Average time of wing and base cmdrs [commanders] is two years. Normally into the third set of value systems, personal taste, experience levels at design start. Project book is pronounced garbage and new preferences are forced on the designer. Normally into fourth set by award. Design is pronounced garbage and change orders ensue. System sucks.

(11) No, based on my experience programming mostly happens during design because so many changes occur during design and even construction. Changes happen because of budgets, change of user (turnover), and lack of engineering (design) experience.

(13) No. However, to integrate programming and design would require organizational and funding approval procedural changes at the headquarters level. The MCP procedures have recently been changed to integrating programming and concept design to a limited degree.

(14) The tools provided by the Air Force and guidance are very flexible. For O&M programming the methods and equipment are adequate but could be improved (mainly by more time to do a good job). Programming MILCON construction is done at to low a level without seeing the big picture and is largely ignored by the Army Corps of Engineers and architect-engineer firms.

(15) Yes and No. The major problem I see is the level of information needed for design cannot be included on the 1391. Also, commander inserts during the design phase has almost certainly produced an interactive programming process.

(17) As a rule they do. There is a time lag usually between the completion of programming documents (i.e. 1391's) and start of design. If the lag time is long, the user has PCS'd and now a new user is ready to explain to the designer what he really wants. In my opinion, programming documents were not intended to be designs. The designer, when doing project books, is often called upon to amplify project requirements and get into the nuts and bolts of the project.

(18) Yes. My determination of whether our programming methods we're using are adequate in defining requirements is to look at the final product - the facility after it is constructed. Our batting average is good. We have excellent facilities. An on-going problem in the military is the changing of user/using agency people during the time between programming and design completion. The long period of time between programming approval and funding of the project usually results in as many as 4 or 5 user/using agency personnel changes. Each has significantly different ideas of need, ect., affecting many design and construction changes. The cost is very significant.

(19) 1. For too much emphasis on making audit trial.

2. Design process (developing total requirements irregardless of mandated approval levels) greatly hindered by too much attention to meeting approval levels.

3. If a building needs repair, renovation, maintenance, or MC, it should be done without the hindrance of project approval levels. Example, if a flat roof needs repair and best method of repair is sloped roof, then a sloped roof should be installed and classified as repair. Now it's MC.

4. All MAJCOM approval levels should be delegated to the base. MC level of approval should be a "floating" level based on the extent of building repair, i.e., if repair costs of a building is $3,000,000, MC level should be $300,000.

(22) No. Lag time between programming and design start often results in using agency personnel changes, thus different/new ideas/requirements.

(23) No. AF regs not current. Because of delays of years between programming and design the functions are independent in most cases. Using agency needs (wants) change as commanders PCS and new ones come in. The AF commanders are determined to get what they (individually) want and nor necessarily what is needed or best in the long run. They have authority but little experience in facility design and in general are very closed minded. [They] Make absolute decisions with very few facts. After programming is set they want more in design with no money to commit. More experienced people needed to man programming and design sections. (This equates to more, high percentage, of positions being civilian not passing through military, i.e. 2Lt, 1Lt, etc. This is my biggest problem and has been for years. Help!

(24) No they do not always define project requirements. MCP, P-341, and P-713 projects are quite well defined through the DD 1391 approval process. There is no definitions for projects within local approval other than value requirements on an AF 332.

(26) Yes.

(27) My answers have been geared to O&M RPMC programming, not MCP. Our programming section does a poor job. Unfortunately they are not under my supervision at this base, which is not smart. The whole delivery process should be under the Chief Engineer's guidance so the programming process could be properly emphasized.

(28) For some projects, yes, for others - no. The structure of any DOD organization results in "changing" personnel. Oftentimes, those "in change" will be two or three different people with different ideas and goals. The concepts change frequently. The programmers and designers have to be "fluid" - but there reaches a point where no matter how thorough your programming documents, new ideas come up during design which alter your original concepts. This is necessarily all bad.

(29) No.

(30) Yes, if you follow the guidelines established. The biggest problem appears to be at HQ USAF or above where people seem to want to tie the hands of the designer to the program document in lieu of accepting the fact that the deeper you dig, the better the solution can be and accept the change.

(31) If the programming section functioned as regulations require it may be satisfactory. However it is generally understaffed with low graded personnel with little or no experience when just the opposite is needed.

(32) Not very well. Key personnel, especially users, change too often between initial programming and actual start of design, especially for MCP projects. Also between design and construction. Also, base programming and design staffs are too small and/or have too many projects to manage especially when you consider that programming and design per se are only one part of the multitude of functions they perform. At bases, there is too much to do and too few personnel to do it. Doing more with less is a cliche that needs to be buried. We can not take the time to do things right and do justice to both the progrrmming and design functions. We hop form one "command interest project" to another.

(33) Our greatest concern still remains adequate manpower to do the job. Air Force needs to get serious about providing the number of people to do a job.

(34) No. The process is much too complex and for the most part the users simply want to say I need this or that and have someone hand them the key to their new facility. They don't want to think, plan, wait, defend or otherwise communicate their needs and justification. The programming process is a mess that wastes manpower. The user should budget and pay for the service "if" he can get the money and approval. We could save manpower by working on only what will be done.

(35) Yes, but requires close interaction between programmers, designers, users, and contract management personnel to assure final product is what user wants.

(36) I believe they can. I think there is an override situation where the users already have it in their mind what they want and what it should cost. They then just want us to provide it. When we're able to get through this situation and ask them about their problems, we usually find out that what they want is not the solution to their problems. Engineers are problem solvers. Tell us your problem and let us use our expertise, experience and creativity to solve your problem.

(38) Yes. However there will also be changes. A design evolves on a major project. The whole philosophy of design is to work out problems/identify problems. This is design development. Contract documents are the final product of this process. If your survey addressed only MILCON or only O&M the answers would be more definite. Also, an expanded definition of programming is needed. You start with your definition of programming which is not inclusive of what the programming process is all about.

(39) In most cases it does. The biggest problem I see is that many times technical requirements are given for equipment that is to be installed, but either during design or after design completion, technical requirements change as equipment is developed. Equipment should be fully developed prior to design so requirements are firm.

General Comments

(38) Questions were answered considering all programs i.e. MILCON, O&M, MFH, NAF, miscl. Programming works differently for each of these programs.

I don't like your definition of programming. Programming is a process of identifying many requirements, verifying the justification for the project, and determination of a priority list to match requirements with available funds. Project definition is the start of the design pha e, usually a 10% design status.

Questions are not worded ideally. For all programming for all programs (MILCON, O&M, MFH, ect.) there must be some degree of thought/discussion of the design concept.

I think there is confusion between programming versus preparation of criteria documents such as project book and RAMP.

Appendix F: Comments From Round Two/Group A Questionnaire

The following are Group A's transcribed written comments and responses from the round two Delphi questionnaire (Appendix C). The comments are organized under the question to which they were responding. Comments not in response to a particular question are found at the end of the appendix, under general comments. The number before each comment is the respondent number, and its only significance is to the researcher for his own records. In addition, bracketed text is added by the researcher, in some cases, to clarify the context of the response.

9. A facility design is a problem solution.

(18) ["Problem solution" is] jargon (CRS). Not illuminating, is much more complex.

(22) I'd like to know what the "E" [strongly disagree] response people say.

10. Programming is the responsibility of the client/owner.

(1) [Programming is the responsibility of the client/owner] with programmer.

(2) Agree, but client/owner needs qualified consultants with specific expertise.

(13) These two questions [10 and 11] are misleading. Ultimately the owner is responsible for the programming. The client/owner may initiate a program, may hire an A-E to perform the programming, and should in good practice client/owner should approve the program and take responsibility for it. As an architect, it is risky to design a project based on a program that the client/owner has not approved in some manner and taken responsibility. The issue of responsibility has a lot of legal and liability implications which should explain the spread of responses (and A-E liability/lawsuit dilemma).

(17) This is the way it is, but not the way it should be.

(18) The comments [on Questions 10 and 11] are more enlightening than the questions.

11. Programming is the responsibility of the designer.

(1) [Programming is the responsibility of the designer] with client.

(13) See Comment on Question 10 above.

(17) See Comment on Question 10 above.

(18) See Comment on Question 10 above.
[Responsibility of designer] only if contracted to do. Design is the responsibility of the designer.

14. Programming is a series of client decisions on the direction of design.

(2) Programming can impact design, but is independent of design process.
(13) Operational decisions have a direct impact on programming.
Programming is usually more of a basis for "planning" than "design".
(17) The client's decisions [important] but are not sole content.

18. Designers should be part of the programming team.

(4) Depends on project.
(18) Depends.

20. It is important to educate the client/users in the architectural design process.

(22) They [clients/users] need help (sometimes) to be wise clients. Most get to do it only once.

22. A facility programming document is primarily information for the designer.

(1) [Facility programming document] also for client.
(2) It is primarily the basis for client/owner decision making and is foundation of project development.
(6) It [programming document] is the basis for a contract.
(13) If "designer" were changed to "planner" or "space planner" I would strongly agree.
In architectural offices, there is a big difference between a designer and a planner, design and planning. (Medical) planners are the primary users of programs.

23. A facility programming document is valuable information for the client.

(2) [See comment for Question 22].
(22) A good program has high client value in several ways: The process is one of organizational self-reflection and self-learning, and often surfaces issues for action ... as well, it provides clear expectations about what they'll get ... and acts as a base for a post-occupancy evaluation.

25. Conceptual Design is part of the programming process.

(2) It [conceptual design] is necessary to test the program.
(4) Conceptual design testing is needed for some projects.
(12) "Is part of" in the sense that design impacts of programming decisions must be assessed.
(22) Ought to be.
(24) [Programming phase is iterative] but only thru "concept" design. Program is tested and refined [with schematics].

26. Programming should be completed prior to design.

(9) All program data cannot be completed prior to any design. Programming data is developed in greater detail in response to design investigation.
(13) Ideally.
(18) Desirable, but can be fast tracked in phases along with design phases.
(22) These [Questions 26 and 27] are compatible, if concept is embedded in program process.

27. Programming should be integrated with conceptual design.

(2) Conceptual design should follow programming.
(4) On some projects.
(22) [See comment on Question 26 above].

28. Programming and conceptual design should be interactive, not separate phases of the facility delivery process.

(13) Practically, or in reality.
(18) Desirable.

29. The programming process is the same for all facility projects.

(15) Basic process, i.e. Data Collection > Data Translation > Report the same but with many variations in techniques, content and format.

31. The end product of programming is information not design.

(1) [The end product is design with] some conceptual design.

35. A programmer or someone on the programming team should have experience in design.

(2) Experience in the project type.

(24) [Someone on programming team should have experience in design [especially if programming/predesign/schematics (concepts) iterate or overlap.

37. A programmer or someone on the programming team should understand the whole building delivery process.

(2) This experience is valuable in anticipating implications.
(22) In detail no. In general yes.
(18) Desirable.

40. During the facility delivery process, the programming - design relationship/connection can be a problem.

(15) Can be but need not be if team participants respect each other's roles and work together.
(22) [Can be] a speculative word ... who could disagree with this wording.

45. Programming is:

A. PART OF THE DESIGN PROCESS
B. SEPARATE FROM THE DESIGN PROCESS
C. INTERACTIVE WITH THE DESIGN PROCESS

(7) Programming + Design = Design Process
Analysis + Synthesis = Design Process
(15) Separate activities, usually with separate staffs, but interactive.
(17) Is [separate from the design process], but should be [interactive with design process].
(18) You gave too narrow a definition of "design" above.

46. Conceptual design is:

A. PART OF THE PROGRAMMING PROCESS
B. PART OF THE DESIGN PROCESS
C. PART OF BOTH THE DESIGN AND PROGRAMMING PROCESSES
D. SEPARATE FROM THE PROGRAMMING AND DESIGN PROCESSES

(1) [Conceptual design is] the link between the two [programming and design processes].
(22) Ought to be [part of the programming process]
Our experience shows that the best programs and highest rate of programming participation and acceptance by clients comes when concept design is used, in the programming process, to test the program and to help the client visualize the implications of a complex program. But this is not "the" conceptual design, only one version ... as a check on the program.

47. The distinct phases of the facility delivery process are:

A. PROGRAMMING, CONCEPTUAL DESIGN, DESIGN, and CONSTRUCTION.
B. PROGRAMMING, DESIGN, and CONSTRUCTION
C. DESIGN and CONSTRUCTION
D. OTHER

(18) [Either answers "A" and "B"] ok.
(22) [Programming, Design, and Construction] only if concept is in program.
(24) I would add "predesign" [to A] which tests basic concept or alternate in many projects with serious site or budget constraints. [I use an iterative approach with overlapping programming phases. Our design phases are Concept, Schematic, Design Development, Construction Documentation.] Iterative during "programming" and "concepts" and to some degree schematic. This is closest to our approach and more realistic in practice. Program must be fixed before entering "design" phase or excessive cost, delays, ect. occur. [Schematics] with program verification and refinement.

48. In your opinion, who should control the programming process of facility projects. (Assume client/owner has no in-house capability and design firm has in-house programming staff).

A. CLIENT/OWNER
B. DESIGNER OR DESIGN TEAM
C. IN-HOUSE PROGRAMMING STAFF (part of the design firm)
D. OUTSIDE PROGRAMMING CONSULTANTS (separate from the design firm)

(2, The independent analysis is essential for objectivity.
(22) Even with definition below, hard to answer. If roles A, B, C, D are available and on-the-job, all will have influence, yes? A weird assumption. So few design firms have other-than-desk-counters as programmers. They have little interest or skill in anything other than functional analysis... Little capacity to examine psyco-social issues or those of organizational culture.

[Outside Programming Consultants] only if client hires them directly. And even then, a client can (and should be able to) reject/negotiate some of the results. So client always retains final "say" on what gets built.
(24) [Client controls] final product and decision, [but in-house staff, designer, and outside consultants should have strong influence] and manage the process.

50. <u>The Architect's Guide to Facility Programming</u> lists three basic approaches to programming, which of the following approaches best describes your programming method.

A. SEGREGATED
B. INTEGRATED
C. INTERACTIVE
D. SEGREGATED OR INTERACTIVE

(13) We perform programming as a separate activity but with key people from the design team taking a lead role.

(17) Typically method I use [segregated]. This [integrated] would be my preferred method [but] context does not permit.

(24) Disagree with this part [programming performed by separate individuals or teams from designers]. I would agree to "A" when eliminated with a predesign activity [to] test program concepts and other pertinent constraints (site, budget).

General Comments

(13) Now that I'm reading these questions a second time and you are getting down to the nitty-gritty, I think you would get completely different results if you substituted planner, space planner, medical planner, ect. for "designer". I am assuming you are using "designer" in a very broad sense.

(22) I sense a problem between descriptive and projective responses. Your use of the word "is" forces me (us) to say what "is" (descriptive of current, general practice). I want to be able to say "what ought to be" (projective) if we were doing it right.

Appendix G: Comments From Round Two/Group B Questionnaire

The following are Group B's transcribed written comments and responses from the round two Delphi questionnaire (Appendix C). The comments are organized under the question to which they were responding. Comments not in response to a particular question are found at the end of the appendix, under general comments. The number before each comment is the respondent number, and its only significance is to the researcher for his own records. In addition, bracketed text is added by the researcher, in some cases, to clarify the context of the response.

7. Facility programming identifies the technical building requirements for design.

(30) Reading [the comments for Question 7] it this way I changed from "D" to "B".
(39) [Only] unique information to project.

8. A facility programming document is a problem definition or statement.

(17) It [programming document] is a funding document that includes a problem definition and one possible solution.

10. Programming is the responsibility of the user/using agency.

(30) Change in user leadership between programming and design, and design and construction make this fruitless. Cross your fingers and hope user doesn't change leadership between design and construction completion.
(32) Confusion is apparent in comments to this question. Programming is not design. They are two separate and distinct entities with different purposes.

14. Programming is a series of user/using agency decisions on the direction of design.

(11) Functional [decisions on direction of design] only.
(23) Ideal situation.
(30) [The user defines the problem] and concurs with acceptable solutions.

[Programming is identifying user requirements and criteria] and formulating a concept of what is needed.

[User should not be making] "technical" [design decisions].

(32) Programming is not design.

18. Designers should be part of the programming team.

(11) [Designers should be] definitely involved prior to finalizing the 1391 documents.

(23) Programmers should have design experience and/or consult appropriate designers in decision making. Can't afford to tie up limited designers on programming that may never get to design.

(27) Not staffed to allow it.

(30) This [designers not available to be part of the programming team] will always be a problem.

(32) Not really necessary. They are different functions.

20. It is important to educate the users/using agencies in architectural design process.

(23) To a limited degree.

(27) [Agree,] so they take programming stage seriously.

(30) Not how to, just what it involves. Right?

There's always a way to make the user part of the team. CE credibility is the first requirement.

21. Three-way communication between the designer, programmer and user/using agency is essential to programming.

(23) If needed by programmer.

(27) Depends on the problem. If programmer can't resolve user's problem, designer should be consulted.

(30) Do it right the first time or do it over - it all takes time. [Referring to comment that lack of designer time is a problem].

(32) No it isn't. In most (if not all) cases, the programmer in not involved at all once design begins. Also, the designer is not even known at the time of programming - i.e. MCP projects.

22. A facility programming document is primarily information for the designer.

(17) [Information] for management and funds.

(30) More than this [justification and budgeting] in reality.

25. Conceptual design is part of the programming process.

(27) Could be. Depends. Not usually done. But could be.

(30) Whether realized or not all programmers go thru a form of conceptual design just to come up with an estimate, so it's only the degree that varies with the needs and manning.

27. Programming should be integrated with conceptual design.

(27) [Agree,] but may not be practical in Air Force system.
(30) Not easy to find "designers" that want to be programmers. The grade is usually lower and so is the prestige. If the Design Section doesn't help and train programmers then that's all you get i.e. a less than satisfactory document.
Could be very tough to expect a programmer to do a RAMP.
(32) You need to separate RPMC form MCP. Actual practice is very different for both.

28. Programming and conceptual design should be interactive, not separate phases of the facility delivery process.

(30) This [programming and design should be interactive] but time between programming and design is often years] the "real" problem especially when noted that using personnel also change.

29. The programming process is the same for all facility projects.

(30) [No] unless you mean how to fill out paperwork.
[Programming] steps [should be the same] but not the details.
(32) RPMC and MCP are quite different.

30. Programming is essential regardless of project size.

(30) In some form yes, but not always to the same detail.
(32) Not true. Frequently, "programming" of RPMC projects is virtually non-existent. Project goes from a line item with a "WAG" directly into design.

31. The end product of programming is information not design.

(17) End product is to obtain funding.
(30) Conceptual design is considered to be information.

34. Programming should always produce a formal document.

(30) On a form but long-hand should be acceptable if neat.

35. A programmer or someone on the programming team should have experience in design.

(23) Absolutely.
(27) Desirable.
(30) Probably will be the exception rather than the norm unless you make it more attractive by the consideration of "conceptual design" versus just filling out documents.
(32) Probably but not necessarily.

40. During the facility delivery process, the programming - design relationship/connection can be a problem.

(30) Chiefs of Design and Programming must communicate regularly. Design Chief must take responsibility to keep programmers informed.

45. Programming is:

A. PART OF THE DESIGN PROCESS.
B. SEPARATE FROM THE DESIGN PROCESS.

(23) Should be [part of the design process]. Actually in Air Force [programming is separate from the design process] mostly due to many years between program/design.

50. <u>The Architect's Guide to Facility Programming</u> lists three basic approaches to programming, in your opinion, which of the following approaches is best.

A. SEGREGATED
B. INTEGRATED
C. INTERACTIVE

(17) We use both. [Use segregated] first, [Use interactive] second.
(27) Do you want best or do you want our process?

53. Which of the following techniques are most effective for ANALYZING and ORGANIZING programming data?

(17) We use all of them [listed techniques] at one time or another.

General Comments

(11) Strong programming sections with inputs from designers for conceptual designs make for better overall projects. Weak programming sections and weak user inputs cause major problems, especially with A/E projects.

(17) The purpose of programming is to advise management of the scope and possible solution to a need. Based upon this management will have sufficient information to approve and fund a project. Design is to produce details of the solution for a contract document or for construction. Programming and design have two different purposes and depending upon management requirements they may be integrated or separate. With programming defined as above all projects will need programming formally or informally. Similarly all projects must have some design. Concept study alternatives, price etc. of different alternatives. I don't think programming should be complicated beyond this definition.

(23) There is no short cut or substitute for experience in either programming or design.

Bibliography

1. Department of the Air Force. Design and Construction Management. AFR 89-1. Washington: HQ USAF, 1 November 1988.

2. Department of the Air Force. Programming Civil Engineer Resources Appropriated Fund Resources. AFR 86-1, Volume I. Washington: HQ USAF, 26 September 1986.

3. Department of the Air Force. Criteria and Standards for Air Force Construction. AFR 88-15. Washington: USAF, December 1985.

4. Department of the Air Force. Project Book (PB) Fromat. Contruction Technical Letter (CTL) 88-1. Washington: HQ USAF/LEEC, Jan 1988.

5. Strategic Air Command. MILCON Process, Requirements and Managment Plan (RAMP), Project Defintion (PD). DEE Operating Instruction 90-1. Offutt AFB: HQ SAC, DCS for Engineer and Services, 15 March 1990.

6. Department of the Air Force. United States Air Force Project Manager's Guide for Design and Construction. Washington: HQ USAF/LEE, June 1989.

7. Dutcher, Capt Gerald B. An Investigation Concerning Perceptions of Military Construction Program Effectiveness by AFCREs, MAHCOMs, and the Bases. MS Thesis, AFIT/GEM/LSM/86S-10. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1986 (AD-A174466).

8. Mogreen, Maj Eric T. The Causes and Costs Of Modifications to Military Construction Constracts. MS Thesis. US Army Command and General Staff College, Fort Leavenworth KS, 1986 (AD-A172833).

9. Stollbrink, Capt Michael. A Study of User Involvement in the Military Construction Program Process. MS Thesis, AFIT/GEM/LSM/86S. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1986 (AD-A174445).

10. HQ USAF/LEE. "Improving the MILCON Process." Briefing Slides. August 1989.

11. U.S. General Accounting Office. Information on the Use of Value Engineering in Federal Design and Construction. Report to the Honorable Nancy Landon Kassebaum, United States Senate; GAO/GGD-85-44. Gaithersburg MD: U.S. General Accounting Office, Document Handling and Information Services Facility, 5 April 1985.

12. Agostoni, Edward J. "Programming: Demanding Specialty In A Complex World," Architectural Record, 93-94 (September 1968).

13. Sanoff, Henery. Methods of Architectural Programming. Stroudsburg PA: Dowden, Hutchinson and Ross, 1977.

14. Gordon, Douglas E. and M. Stephanie Stubbs. "Programming," Architecture, 203-210 (May 1988).

15. Pena, William M. and William W. Caudill. "Architectural Analysis - Prelude to Good Design," Architectural Record, 178-182 (May 1959).

16. White, Edward T. Interviews with Architects About Facility Programming. Tallahassee FL: Graduate Program, School of Architecture, Florida A&M University, 1982.

17. Palmer Mickey A. The Architect's Guide to Facility Programming. New York: McGraw-Hill Book Company, 1981.

18. White, Edward T. Introduction to Architectural Programming. Tucson AZ: Architectural Media, 1972.

19. AIA Practice Committee, Sub-Committee on "The Programming Process." 3 April 1990.

20. Wade, John W. Architecture, Problems, and Purposes: Architectural Design As A Problem-Solving Process. New York: John Wiley and Sons, 1977.

21. Pena, William and others. Problem Seeking: An Architectural Programming Primer (Third Edition). Washington: AIA Press, 1987.

22. Haviland, David. "The Building Enterprise," The Architect's Hanbook of Professional Practice (Eleventh Edition), Volume 1. Washington: The American Institute of Architects, 1988.

23. Haviland, David. "Predesign Services," The Architect's Hanbook of Professional Practice (Eleventh Edition), Volume 2. Washington: The American Institute of Architects, 1987.

25. Kemper, Alfred M. Architectural Handbook: Environmental Analysis, Architectural Programming, Design and Technology, and Construction. New York: John Wiley and Sons, 1979.

24. Kliment, Stephen A. "Professional Options," The Architect's Hanbook of Professional Practice (Eleventh Edition), Volume 1. Washington: The American Institute of Architects, 1988.

26. Davis, Gerald. "How To Buy A Better Building," Business Horizons, 25-32 (February 1968).

27. Wade, John W. "Architectural Programming," Introduction to Architecture, edited by James C. Snyder and Anthony J. Catanese. New York: McGraw-Hill Book Company, 1979.

28. Monroe, Linda. "Committing to the Computer Game," Buildings, 68-72 (May 1989).

29. FM:Space-Management. FM:Systems, Raleigh NC, undated.

30. White, Edward T. Project Programming: A Growing Architectural Service. Tallahassee FL: Graduate Program, School of Architecture, Florida A&M University, 1982.

31. Sackman, H. Delphi Assessment: Expert Opinion, Forecasting, and Group Process. RAND Report R-1288-PR. Santa Monica CA: The RAND Corporation, April 1984 (AD-786878).

32. Dalkey, Norman C. Delphi. Rand Report P-3704. Santa Monica CA: The RAND Corporation, October 1967 (AD-660554).

33. Brown, Bernice B. Delphi Process: A Methodology Used for the Elicitation of Opinions of Experts. RAND Report P-3925. Santa Monica CA: The RAND Corporation, September 1968 (AD-675981).

34. Emory, C. William. Business Research Methods (Third Edition). Homewood IL: Irwin, 1985.

35. Devore, Jay L. Probability and Statistics for Engineering and the Sciences (Second Edition). Monterey CA: Brooks/Cole Publishing Company, 1987.

Vita

Captain Michael A. Ross [redacted] He, however, grew up in Northbrook, Illinois. He graduated from Notre Dame High School in Niles, Illinois in 1978 and attended the University of Illinois at Urbana-Champaign, graduating with a Bachelor of Science in Architectural Studies in May 1982. Upon graduation, he worked as an intern-architect for Carol Ross Barney, Architects in Chicago. His chief accomplishment, during this time, was the development of a programming document for a maritime museum. In April 1983, he entered Officer Training School, Lackland Air Force Base, Texas and received his commission. His first assignment was to Moody Air Force Base, Georgia as the base architect. Here, he worked primarily as a designer and project manager on more than eighty construction projects. After almost five years, he left to serve in Saudia Arabia as the project construction manager for four radar sites. The radar sites were built for the Royal Saudia Air Force as part of Peace Shield, a Foreign Military Sales program. Upon completing this assignment, he entered the School of Systems and Logistics, Air Force Institute of Technology, in May 1989.

Permanent Address: 601 Linden
Wilmette, Illinois 60091

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1990	Master's Thesis

4. TITLE AND SUBTITLE
AN ANALYSIS OF AIR FORCE FACILITY PROGRAMMING AND ITS EFFECT ON DESIGN AND CONSTRUCTION

5. FUNDING NUMBERS

6. AUTHOR(S)
Michael A. Ross, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GEM/DEE/90S-14

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)* A key component of Air Force Civil Engineering project management is facility programming, the identification of requirements for construction projects. The literature review revealed that inadequate identification of facility requirements has lead to unsatisfied facility users, excessive cost growth, rework of construction documents, loss of projects, and change orders during construction. The goal of the research was to identify potential improvements to the programming processes used by the Air Force. The "Delphi Technique" was used to solicit information about programming from two panels of "experts": (1) chief engineers within Base Civil Engineering organizations, and (2) professional programmers outside the Air Force. The respondents answered questions about programming in two rounds of questionnaires. Comparisons were made between the groups about current practices and attitudes about programming. The research uncovered significant differences between how the two groups view and use facility programming. From the conclusions, the researcher proposed a new programming model that solves some current problems, and takes advantage of "good" programming practices. The key features are that programming and conceptual design are interactive processes, and the emphasis on functional programming.

14. SUBJECT TERMS
Facility Programming
Project Requirements
Civil Engineering

15. NUMBER OF PAGES
308

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102